SCENE SEGMENTATION USING SIMILARITY, MOTION AND DEPTH BASED CUES

by

Bhargav Kumar Mitra

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY AT THE UNIVERSITY OF SUSSEX

School of Engineering and Design

University of Sussex

Brighton

June 2010

Statement

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature

Bhargav Kumar Mitra
Dated: 23 June 2010

UNIVERSITY OF SUSSEX

BHARGAV KUMAR MITRA; SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCENE SEGMENTATION USING SIMILARITY, MOTION AND DEPTH BASED CUES

Summary

Segmentation of complex scenes to aid surveillance is still considered an open research problem. In this thesis a computational model (CM) has been developed to classify a scene into foreground, moving-shadow and background regions. It has been demonstrated how the CM, with the optional use of a channel ratio test, can be applied to demarcate foreground shadow regions in indoor scenes illuminated by a fixed incandescent source of light.
A combined approach, involving the CM working in tandem with a traditional motion cue based segmentation method, has also been constructed. In the combined approach, the CM is applied to segregate the foreground shaded regions in a current frame based on a binary mask generated using a standard background subtraction process (BSP). Various popular outlier detection strategies have been investigated to assess their suitabilities in generating a threshold automatically, required to develop a binary mask from a difference frame, the outcome of the BSP.
To evaluate the full scope of the pixel labeling capabilities of the CM and to estimate the associated time constraints, the model is deployed for foreground scene segmentation in recorded real-life video streams. The observations made validate the satisfactory performance of the model in most cases.

In the second part of the thesis depth based cues have been exploited to perform the task of foreground scene segmentation. An active structured light based depth-estimating arrangement has been modeled in the thesis; the choice of modeling an active system over a passive stereovision one has been made to alleviate some of the difficulties associated with the classical correspondence problem. The model developed not only facilitates use of the set-up but also makes possible a method to increase the working volume of the system without explicitly encoding the projected structured pattern.
Finally, it is explained how scene segmentation can be accomplished based solely on the structured pattern disparity information, without generating explicit depth-maps. To de-noise the difference frames, generated using the developed method, two median filtering schemes have been implemented. The working of one of the schemes is advocated for practical use and is described in terms of discrete morphological operators, thus facilitating hardware realisation of the method to speed-up the de-noising process.

Dedicated to Ma and Baba with love

Acknowledgements

A deep sense of indebtedness impels me to put on record a note of gratitude to all who consciously or unconsciously have made contributions to my research pursuit. I would like to start by first thanking Dr. Philip Birch (Phil), my main supervisor, for all the unstinted support and advice rendered to me by him over the last couple of years. Apart from being an inspirational supervisor, Phil has been a friend and mentor. I have been able to gain deep insights into my research topic and hone my programming skills from the frequent discussions we have had either in his office or in that of mine. It is because of Phil that I now know many of the tricks of the trade and am confident of undertaking more novel and cutting-edge research work in future.

A special thanks is due to my co-supervisor, Dr. Rupert Young. Rupert bears an affable and charming personality and is known in our group for his deep and sound knowledge on various Fourier based machine-vision methods. I barge in his office, almost everyday in the evening, for a mind-boggling discussion over a wide range of topics. It is from these discussions, I have realised that a physical understanding of a problem is vital to conduct research in any specific technical field. In other words, a general understanding of a problem helps in getting to the pulp of the conundrum in an efficient way, without getting lost in crazy mathematics — a helpful tip to comprehend what has been said in a research paper. Rupert also played an instrumental role in arranging funding for my doctoral studies and helped me sort out various administration related problems. I thank him again for the kind cooperation that he bestowed upon me in all matters for the asking. Everytime we meet, he greets me with a smile on his face which is greatly soothing, and at once dispels any hesitation on my part for approaching him with any problem. I literally owe my DPhil to him.

I would also like to take this opportunity to tender my acknowledgement of indebtedness to my other co-supervisor, Prof. Chris Chatwin. Chris is the director of our research group and has long years of experience in research on various fields with numerous publications to his credit. I have had the honour to work under him and receive valuable advice and directions regarding strategies to be adopted in facing challenging research problems. He made me understand that emotional balance is necessary for a successful research career in the practical world. I owe a deep sense of gratitude to him for making my DPhil a truly productive and stimulating one.

I am thankful to the University of Sussex for providing me with a three year DPhil Bursary and a Graduate Teaching Assistantship for completing my doctoral studies, and Spiral Scratch Limited, UK for providing us with a range-camera set-up on which a major part of the thesis is based.

Undertaking doctoral studies at a foreign university, particularly in the UK, was one of my long-cherished dreams. The fact that I have been able to materialise it would not have happened, but for the sheer sacrifices made by parents. It is because of their constant encouragement and blessings that I have been able to complete my doctoral studies and will be looking forward to pursuing a successful career as a researcher. I cannot think of a way of returning them the favour, be-

cause whatever I will do, will be too small compared to what they have done for me.

I would be called ungrateful if I do not thank the wonderful administration and support staff at the School Office (specially Alan, Richard, Anita and Linda); it is because of them that my DPhil has been, by and large, a plain sail. Thanks also go to my research colleagues and students with whom I have spent a wonderful time at Sussex.

Finally, I would like to thank some of my fantastic friends: Rafael, Christina, Helena, Fiona, Joel, Anna, Daniela, Mike, Clare, Yumiko, Prosenjit, Biswajoy, Shouvik and Priya for always standing by me during the ups and downs of my life. I am also grateful to Dr. Abhijit Mitra of IIT Guwahati, India for motivating me to undertake doctoral studies and inducing within me an urge to conduct quality research and publish papers.

Bhargav Mitra
University of Sussex
May 2010.

Arise, awake and stop not till the goal is reached.
Swami Vivekananda (1863 – 1902)

A mind all logic is like a knife all blade. It makes the hand bleed that uses it.
Rabindranath Tagore (1861 – 1941)

If you shut the door to all errors, truth will be shut out.
Rabindranath Tagore (1861 – 1941)

Never accept an idea as long as you, yourself, are not satisfied with its consistency and logical structure on which the concepts are based. Study the Masters. These are the people who have made significant contributions to the subject. Lesser authorities clearly bypass the difficult points.
Satyendra Nath Bose (1894 - 1974)

Contents

List of Figures

List of tables

Chapter 2

Chapter 3

Chapter 5

Publications resulting from thesis

JOURNAL PUBLICATIONS:

[1] Philip Birch, **Bhargav Kumar Mitra**, Nagachetan Bangalore, Saad Rehman, Rupert Young, Chris Chatwin, "Approximate bandpass and frequency response models of the difference of Gaussian filter ", in press, *Optics Communications*, 2010.

[2] **Bhargav Kumar Mitra**, Philip Birch, Rupert Young, Chris Chatwin, "Efficient de-noising of images using a non-aggressive median filtering scheme", Under Review, *Optical Engineering*, 2010.

[3] **Bhargav Kumar Mitra**, Nagachetan Bangalore, Waqas Hassan, Philip Birch, Rupert Young, Chris Chatwin, "Change-of-illumination tolerant scene surveillance using a multi-stage edge detector", in press, *Asian Journal of Physics*, 2010.

[4] **Bhargav Kumar Mitra**, Rupert Young, Chris Chatwin,"On shadow elimination after moving region segmentation based on different threshold selection strategies", *Optics and Lasers in Engineering*, vol. 45, no. 11, pp. 1088 − 1093, July 2007.

CONFERENCE AND SEMINAR PUBLICATIONS:

[1] Waqas Hassan, **Bhargav Kumar Mitra**, Philip Birch, Rupert Young, Chris Chatwin, "Human Intrusion Detection Using Robust Histogram Matching Techniques", submitted to *SPIE Defense, Security + Sensing*, April 2011.

[2] **Bhargav Kumar Mitra**, Waqas Hassan, Philip Birch, Akber Gardezi, Rupert Young, Chris Chatwin, "A Two-Stage Approach to Detect Abandoned Baggage in Public Places", in *Proc. of Visual Information Processing XIX, SPIE Defense, Security + Sensing*, vol. 7701, Orlando, Florida, USA, 5 − 9 April, 2010, pp. 770107-1:8.

[3] Waqas Hassan, **Bhargav Kumar Mitra**, Philip Birch, Rupert Young, Chris Chatwin, "Illumination Invariant Method to Identify and Track Abandoned Objects in Public Places", in *Proc. of Optical Pattern Recognition XXI, SPIE Defense, Security + Sensing*, vol. 7696, Orlando, Florida, USA, 5−9 April, 2010, pp. 76961V-1:9.

[4] Waqas Hassan, **Bhargav Kumar Mitra**, Nagachetan Bangalore, Philip Birch, Rupert Young, Chris Chatwin, "Image processing methods for event detection from video surveillance sequences", in press, *Information Technologies, Systems and Networks (ITSN* - 2010*)*, Moldova, May 2010.

[5] Chris Chatwin, Rupert Young, Philip Birch, Waqas Hassan, **Bhargav Kumar Mitra**, Nagachetan Bangalore, Ioannis Kypraios, "Global Panopticon", *IET*, Invited Presentation, The Hawth-Spotlight, Crawley, Sussex, UK, 8^{th} October, 2009.

[6] **Bhargav Kumar Mitra**, Muhammad Kamran Fiaz, Ioannis Kypraios, Philip Birch, Rupert Young, Chris Chatwin, "Performance Analysis of a Modified Moving Shadow Elimination Method Developed for Indoor Scene Activity Tracking", in *Proc. of SPIE Security+Defense- Optics and Photonics for Counterterrorism and Crime Fighting IV*, vol. 7119, Cardiff, Wales, United Kingdom, $15 - 16$ September, 2008, pp. $71190A - 1 : 10$.

[7] **Bhargav Kumar Mitra**, Philip Birch, Ioannis Kypraios, Rupert Young, Chris Chatwin, "On a Method to Eliminate Moving Shadows from Video Sequences", in *Proc. of SPIE Photonics Europe- Optical and Digital Image Processing*, vol. 7000, Strasbourg, France, $7 - 10$ April, 2008, pp. $700012 - 1 : 9$.

PAPERS IN PREPARATION:

[1] **Bhargav Kumar Mitra**, Philip Birch, Waqas Hassan, Rupert Young, Chris Chatwin, "A Method to Detect and Eliminate False-Targets in Surveillance Videos".

[2] **Bhargav Kumar Mitra**, Philip Birch, Rupert Young, Chris Chatwin, "Working Volume Improvement of a Structured Light Based Range Camera".

[3] **Bhargav Kumar Mitra**, Philip Birch, Rupert Young, Chris Chatwin, "De-noising depth cue based segmented scenes using custom-built rank order filtering schemes".

Chapter 1

INTRODUCTION

Scene segmentation, perhaps, is the most crucial step to properly analyse the contents of an image. The step is so critical that the complexity of subsequent processing and eventual success of reaching the desired solution depends, by and large, solely upon it. In fact, Gonzalez and Woods in their book 'Digital Image Processing' [1] have clearly mentioned that an effective subdivision of an image into its constituent parts rarely fails to lead to a successful solution of the problem at hand. However, the irony is that development and implementation of an autonomous and efficient scene classification method is one of the most difficult tasks in the field of digital image processing. Consider a 2-D bi-valued sequence shown in Fig. 1.1(a). The pixels in the background have a flat intensity value of 128 and those inside the box a constant value of 250. In such an image the box inside can be easily segmented out by constructing a pixel gray-value histogram and choosing a value in the trough region of the bimodal distribution as a threshold to segment the image. However, scenes in reality are rarely this simple. Think of a monochrome image of a busy tube station (Fig. 1.2)[†], where the task is to segment the passengers standing on the platform waiting for a train to arrive and the complexity and the difficulty of the problem becomes evident. Traditionally automatic methods to accomplish the objective of demarcating objects of interest in a scene to prepare for high-level feature based processing can be broadly classified into the following categories: (a)

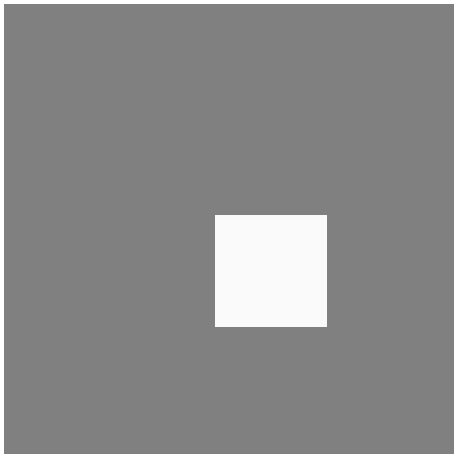
[†]The image has been taken from the Imagery Library for Intelligent Detection Systems (iLIDS) dataset produced by the Home Office Scientific Development Branch in partnership with Security Services, United Kingdom.

discontinuity based segmentation; (b) similarity based segmentation; (c) segmentation using motion cues; and (d) segmentation based on depth estimates. Each of these different automatic scene segmentation methods is briefly discussed in the following paragraphs.

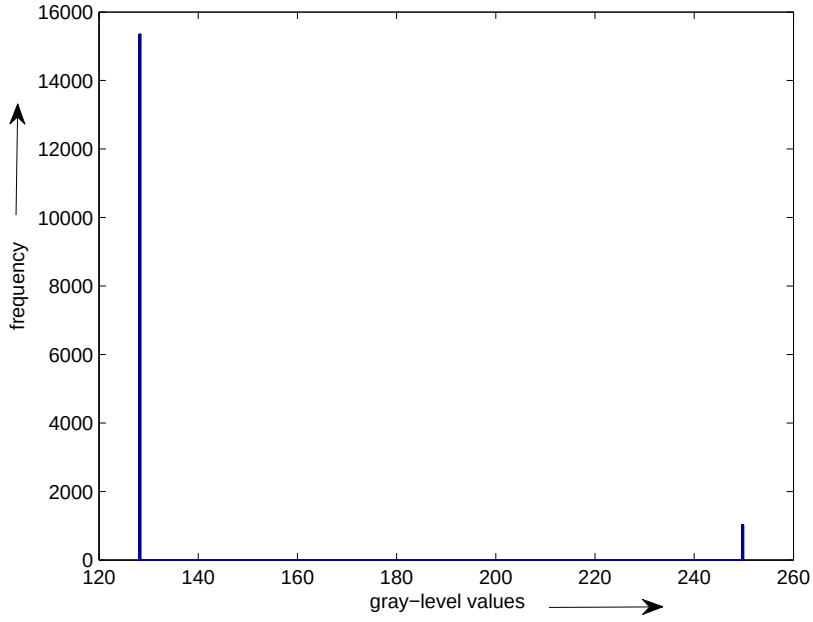
Detection of pixel intensity discontinuities in monochrome images is usually achieved through the use of edge detectors and edge-linking algorithms [1], [2]. However, it should be noted that presence of weak features (edges) in an image and noise, more or less, stymies the working performance of any edge detector [1], [2], [3], [4] and edge linking algorithms are usually heuristic in nature [1], [3]. Moreover, edge detectors also segment shaded regions in an image, usually, as separate regions of interest; in many applications this is regarded as a misclassification step necessitating additional processing measures to be undertaken to ignore the mis-segmented regions [1]. Note here that a method originally developed to detect edges in monochrome images can also be applied in colour images through the deployment of the same detection scheme in each of the bands of the colour image [2]. However, applying an edge detecting scheme to each of the colour bands of a colour image calls for the subsequent requirement of an edge linking algorithm. How edge maps of each colour band can be linked in multiband images and other methods of segmenting regions of interest in multiband/multispectral images have been extensively discussed in [2].

A thorough literature survey reveals a plethora of methods based on similarity to classify regions in an image. These methods form clusters by grouping pixels in an image either based on some basic similarity measure like intensity or association among pixel-predicates that may be derived using the spectral, temporal and spatial properties of each of the pixels in the image.

If the objective is to segment moving foreground objects from a background then motion cues can be effectively used to perform the task of classification [1]. In general, motion based segmentation can be further sub-categorised under two broad classes: spatial domain techniques and frequency domain techniques. An early work on motion based segmentation in the spatial domain can be found in [5]; for a comprehensive study of Fourier based methods to perform segmentation and measure



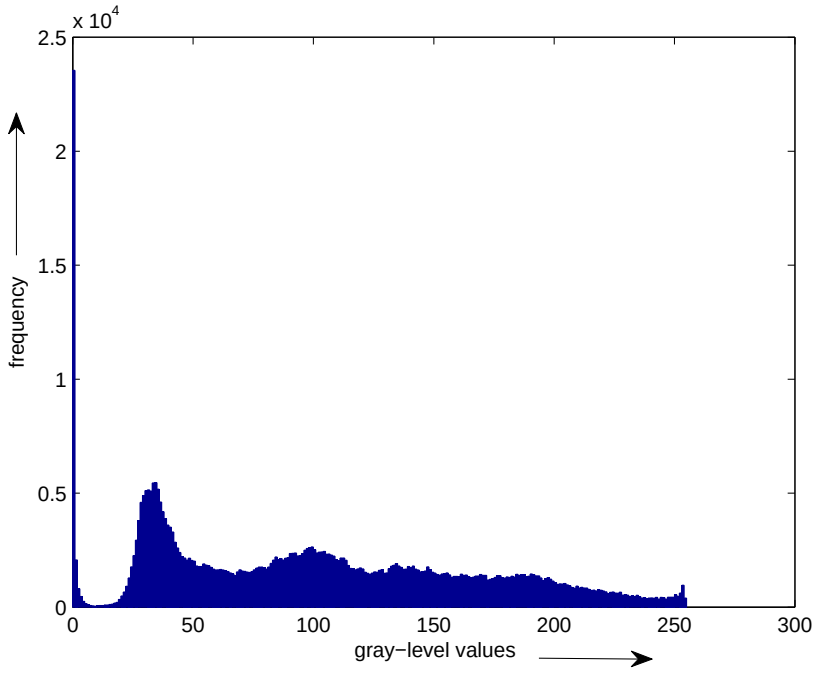
(a)



(b)

Figure 1.1: (a) A 2−D bivalued sequence — the background has got a flat intensity value of 128 and the box inside a constant value of 250; (b) the histogram constructed using the pixel gray-level values of the sequence shows that the inner box can be demarcated by choosing a single gray-level value from the trough region of the bi-modal distribution.

(a)



(b)

Figure 1.2: (a) A London underground tube station scene; (b) the histogram constructed using the pixel gray-level values of the complex scene shows that it is impossible to isolate the passengers in the scene using a straightforward histogram-thresholding based segmentation approach.

velocity, readers are referred to [6]. Note that in this thesis, in future, whenever any reference to any motion based segmentation technique will be made, it will implicitly mean that the technique belongs to the spatial domain class.
A variety of depth estimation methods have been developed over the last four decades for a range of applications. These have lead to the formation of another robust method of effectively segregating multiple partially overlapping or non overlapping foreground objects from the remainder of the scene. However, it has to kept in mind that applying a range estimate based method for foreground scene isolation brings with it additional requirements and/or constraints over the environment under view, as will be explained later in this chapter.

All these broad mechanisms to segment regions of interest in an image can be grouped together as candidate members of the 'bottom-up approach' set of image analysis [7]. A bottom-up approach is usually deployed if no well defined knowledge of an object's features are available *a-priori*. The general content of any arbitrary image is considered while using a particular technique belonging to the set. Structured accessories and methods are deployed to determine standard features, if existing, in an image. Once parts of an image are demarcated using the determined standard features, some or all of the parts, separated, are processed individually to extract more high level features that can be subsequently used for further analysis and/or recognition. In contrast, in a top-down approach model [7] it is considered that well defined feature sets (templates) of the objects to be segregated from an image are available *a-priori*. Regions of an image that match with one of these defined templates are then segmented for further analysis, recognition or detection. Correlation pattern recognition methods [8], [9], [10] [11], [12], [13], [14], [15] are used, in general, to perform the task of template matching. Deformable templates are also used in case of non-rigid object segmentation [16]. Note that an edge map generation process can also be treated as a correlation based feature recognition method [17]; however, deployment of an edge map based segmentation will help in identifying only standard features, if present, in an image. In cases where high-level (object specific) feature sets are available a top-down approach is generally preferred over a bottom-up one.

1.1 Foreground object segmentation using computational models and motion cues

Models and methods developed in this thesis are mainly directed to segment foreground objects from surrounding static or slowly changing complex background scenes. A thorough literature survey reveals that, traditionally, motion cues have been exploited to meet this requirement [1], [18], [19], [20], [21]. The general framework adopted in such approaches involves subtracting the current frame from a static background frame or an expected background frame generated using a temporal averaging process. One of the major drawbacks of this approach is that shadows cast by moving objects get included in the difference frame or the mask generated using it; this leads to misclassification of foreground objects and can severely affect subsequent object detection and tracking processes. Though detection of shadows is sometimes desired and hence required for some applications [21], [22], for video surveillance problems, in general, it is treated as a mis-segmentation [22]. Note that shadows cast by static or dynamic objects are sometimes detected and analysed to infer geometric properties of the objects ('shape from shadow' approaches) [23], [24], [25], [22] or to extract more information about the scene, such as light source direction detection [26], [22]. Since the main objective of this thesis is to segment foreground objects from the remainder of a frame to perform scene surveillance, shadows cast by moving objects have been strictly treated as false-targets; models and methods have been developed in Chapters 2 and 3 to demarcate and eliminate all moving shaded segmented regions to avoid false-target detection and tracking.

Since the working of a motion based based segmentation method is stymied due to the fact that it also segregates moving shadow regions along with the real-targets in the background, a computational model that works in the Red Green Blue (RGB) colour space has been developed in Chapter 2 that labels and eliminates all moving shaded regions in a scene. The computational model developed is analogous to that described by Horprasert *et. al.* [21] and exploits the fact that a 'shadow is a semi-transparent region that retains a representation of the underlying surface pattern, texture or colour value' [21], [27]. The model developed constructs a colour vector

in the RGB colour space for every pixel in the current frame and then decomposes it to separate the chromaticity factor, represented by the vector, from its brightness content. Both the estimates for the i^{th} pixel in the current frame are then compared with those of the i^{th} pixel of the expected background frame to classify a pixel either as a foreground object pixel, a foreground shadow pixel or a background pixel. Note in Chapter 2 it has been shown how the developed model can be used to mark and eliminate only the foreground shadow pixels. The colour model developed by Horprasert *et. al.* in [21] also decomposes a colour vector into its chromaticity and brightness components; however their way of estimating the two components and their pixel classification rules are different from the method employed by the computational model developed in this thesis. Moreover, the model developed in [21] has got two pitfalls that not only makes the background update process complex but also the overall process computationally intensive.

The model attempts to balance a pixel's colour bands by normalising those by their corresponding variance estimates calculated over N background frames. This attempt necessitates additional precautionary measures to be taken as the variance of saturated pixels are usually quite close to zero. In addition, the chromaticity and brightness distortions calculated for each pixel are also normalised using their individual root mean square (rms) values. Minimum estimates for each of the distortions require pre-estimation to tackle situations where the distortion estimates overshoot during the normalising process because of close to zero individual rms values. Though measures that need to be taken to speed-up the process have been mentioned in the paper [21], it is not hard to deduce that the method will work well only on indoor scenes where illumination change is limited. The findings made by Prati *et. al.* [22], independently, confirms the fact that the model works well on indoor scenes as compared to outdoor ones. In comparison the model developed in Chapter 2 is computationally less intensive and consequently makes the background update process fairly straightforward. Moreover, while developing the computational model in Chapter 2 additional attention has been given to indoor scenes illuminated by a single incandescent source of light, a scene that has not received proper attention in the literature. The efficiency of the model has also been tested by deploying it in a real-life application requiring real-time processing of video

frames. The scenario, and how the model has been applied, have been described in Chapter 4. Note that in Chapter 4 the model has been applied to segment the real foreground objects from a complex scene ignoring both moving shadows and other background objects.

It should be mentioned here that other than the model described in [21] and the model developed in Chapter 2, many more statistical parametric or non-parametric models have been proposed in the past few years to detect and eliminate foreground shadows in a scene; a few important methods have been described and analysed in [22].

Deterministic models have also been developed over the last two decades to address the same issue of foreground object shadow removal from video scenes [28], [29], [22]. A thorough physics based model has been developed by Stauder *et. al.* [28]. This model has served as the basic model for most of the deterministic and statistical models developed thereafter [22]. In their paper [28], the authors have described the appearance of a cast shadow in a video-camera viewed scene using an image signal model. Then, based on four assumptions, three criteria have been formed, the combination of the results of which indicate the regions being changed by moving cast shadows in succeeding images. Though the approach developed is, perhaps, the most robust and thorough deterministic method of moving shadow detection, implementation of it suffers due to a number of facts. Firstly, the assumptions made, though practical, lack in generality. Secondly, the approach developed depends on the detection of static and moving background edges, a process which is considered not easy, by any means, and prone to errors. Thirdly, a faithful implementation of the approach developed requires applying it on every succeeding frame; this makes the model sensitive to noise. Finally, the computational load of the process is heavy and this prohibits the use of it in any real-time applications; this has also been pointed out by authors in [22]. Keeping all these factors in mind any attempt to implement this deterministic model has been avoided in this thesis.

Since motion can be used in a very fast and effective way to segment foreground

object pixels from the expected background, a modified method is developed in Chapter 3 of this thesis that uses both motion segmentation and the developed computational method to extract real targets from the remainder of the scene. Use of the computational model eliminates the shortcomings of a motion cue based segmentation process by effectively marking the shaded regions in the mask generated after subtracting the current frame from the expected background frame. On the other hand, use of the background subtraction method not only reduces the search area for the, otherwise pixel-wise applied computational method, but also makes the relaxation of the thresholds used in the computational model possible. Relaxation of the thresholds of the computational model helps in detecting both the soft and strong portions of the moving cast shadows in the target scenes.

It has also been noted that the background subtraction method of moving region separation in the frames falters because of the fact that noise creeps into the binary mask generated from the difference frame due to improper choice of a threshold. Moreover, the issue that this threshold selection process should be free from human intervention adds to the complexity of the problem. To address this issue some popular outlier detection strategies [30], [31], [32], [33] generally used in process control mechanisms, are studied and implemented in Chapter 3 to assess their suitability in difference-frame-to-binary-mask-generation-through-autonomous-selection-of-a-threshold applications.

Discussions on foreground scene segmentation or efficient ways of background estimation is incomplete without the brief mention of two other methods that have received considerable attention from the scene surveillance research community. Foreground scene isolation can be achieved by modeling pixel-predicates using statistical distributions [34], [35], [36] or through the use of a background subtraction process where the background is updated based on an (adaptive) learning rate [37]. However, it should be noted that the computational intensity of these methods sometimes make them unsuitable for real-time applications. Moreover, these methods, in general, do not adequately address the shadow-problem, thereby necessitating further processing to avoid false-target detection and tracking.

Foreground object segmentation in a complex scene can also be achieved through

the use of range estimates i.e. finding out how far the objects in a scene are from a camera (in a co-ordinate system) or through the determination of disparity in some projected pattern. Though such segmentation methods bring with them additional requirements, they are usually known to be robust and effective in indoor scenes or places with a controlled environment. One such range estimating arrangement has been modeled in Chapter 5 and an effective way of segmenting foreground objects through the use of spot position disparity information outlined in Chapter 6. To facilitate understanding of the model developed in Chapter 5, a preliminary knowledge of vision based depth perception theory is necessary. The following section of this chapter fulfills this requirement.

1.2 Generalised stereovision system

A common stereovision system contains two (or more) optical sensors stationed at fixed locations; thus the relative positions of the sensors remain stationary at any point of time. The sensors are usually assumed to be pinhole cameras [38], [39] while modeling the working of the arrangement. Though this is not in fact true, as an optical sensor carries a system of lenses and the distance between the lens-centre and the image plane does not remain constant, the assumption is made to keep the modeling process straightforward. Moreover, if the object distance from the principal point of the lens is relatively much greater than the focal length of the lens system, then the distance between the lens-centre and the image plane remains constant for any projection [38], [39], [40]. Sometimes, also for the sake of convenience, it is assumed that instead of the lens system a single thin lens is viewing the scene under consideration. Here also, it has to be ensured that the object distance is considerably more than the estimated focal length of the assumed thin lens. Moreover, in both the situations, the lens system is calibrated using available camera calibration techniques [41], [42] and the calibration results are in turn used to correct images (compensate for distortion) viewed through it.

Consider a stereovision arrangement (Fig. 1.3) made up of two optical sensors, OS_1 and OS_2, with overlapping footprints, placed at fixed locations and a computer to do the required processing. A general co-ordinate system with centre, O_{π_1}, and axes

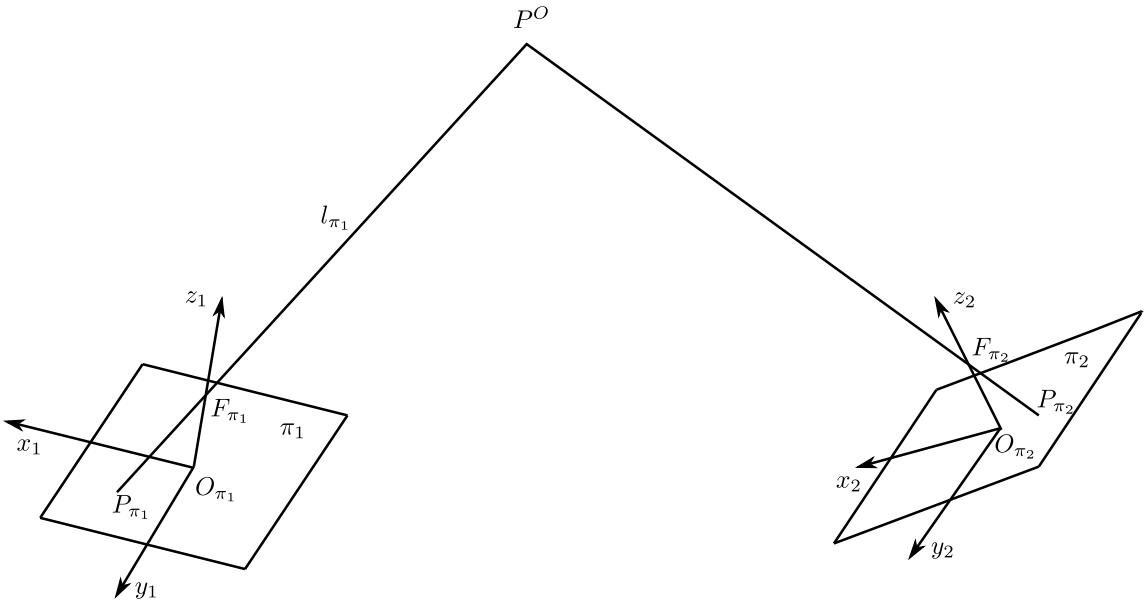


Figure 1.3: A general stereovision arrangement made up of two sensors.

(x_1, y_1, z_1) is fixed at the centre of the image plane, $\langle \pi_1 \rangle$, of OS_1. The z_1-axis of the co-ordinate system is coincident with the optical axis of the sensor and the image plane, $\langle \pi_1 \rangle$, with its $x_1 - y_1$ plane. A light ray from any object point in the field of view of the sensor passes through the focal point, F_{π_1}, of the sensor situated on the z_1-axis of the co-ordinate system and impinges on its image plane, $\langle \pi_1 \rangle$. Let, P^O, be one such object point, a light ray from which passes through F_{π_1} and has a projection P_{π_1} on $\langle \pi_1 \rangle$. Let the co-ordinates of P^O and P_{π_1} with respect to the general co-ordinate system, be $(x^{O_1}, y^{O_1}, z^{O_1})$ and $(x_{\pi_1}, y_{\pi_1}, 0)$ respectively. Since all object points within the range of the sensor are projected through the focal point of the sensor system, P^O and P_{π_1} can be related as follows:

$$P_{\pi_1} = F_{\pi_1} + p_1(P^O - F_{\pi_1}) \tag{1.1}$$

where, in equation (1.1), p_1 is the proportionality constant.
This can be expressed in matrix form as follows:

$$\begin{bmatrix} x_{\pi_1} \\ y_{\pi_1} \\ z_{\pi_1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_1 \end{bmatrix} + p_1 \begin{bmatrix} x^{O_1} \\ y^{O_1} \\ z^{O_1} - f_1 \end{bmatrix} \tag{1.2}$$

where, in equation (1.2), $f_1 = \overline{O_{\pi_1} F_{\pi_1}}$.

Equation (1.2) simplifies to:

$$\frac{x^{O_1}}{x_{\pi_1}} = \frac{y^{O_1}}{y_{\pi_1}} = \frac{f_1 - z^{O_1}}{f_1} \tag{1.3}$$

Similarly, consider another co-ordinate system with centre, O_{π_2}, and axes (x_2, y_2, z_2) fixed at the centre of the image plane, $\langle \pi_2 \rangle$, of the optical sensor, OS_2. As before it is assumed that the optical axis of the sensor coincides with the z_2-axis of the co-ordinate system and that the sensor's image plane lies on its $x_2 - y_2$ plane. The second co-ordinate system can be related with the first one through a 3×3 orthonormal matrix $\mathbf{R}$ describing rotation and a 3×1 column vector, $\mathbf{T}$, describing translation, in the following way:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \mathbf{T} \tag{1.4}$$

Now, the same point P^O $\left[\equiv (x^{O_2}, y^{O_2}, z^{O_2})\right.$ with respect to $\left.O_{\pi_2}\right]$ is imaged by the sensor OS_2 via its focal point, F_{π_2}. Let the projection of P^O on the image plane of OS_2, $\langle \pi_2 \rangle$, be P_{π_2} $[\equiv (x_{\pi_2}, y_{\pi_2}, 0)]$. Again, as before P_{π_2} and P^O can be related by:

$$P_{\pi_2} = F_{\pi_2} + p_2 (P^O - F_{\pi_2}) \tag{1.5}$$

which if expanded in matrix form gives us:

$$\begin{bmatrix} x_{\pi_2} \\ y_{\pi_2} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_{\pi_2} \end{bmatrix} + p_2 \begin{bmatrix} x^{O_2} \\ y^{O_2} \\ z^{O_2} - f_2 \end{bmatrix} \tag{1.6}$$

where, in equation (1.6), $f_2 = \overline{O_{\pi_2} F_{\pi_2}}$.

Equation (1.6) can be further simplified into:

$$\frac{x^{O_2}}{x_{\pi_2}} = \frac{y^{O_2}}{y_{\pi_2}} = \frac{f_2 - z^{O_2}}{f_2} \tag{1.7}$$

If the elements of $\mathbf{R}$ and $\mathbf{T}$ are *a priori* known to us (through a calibration process), then the 3−D co-ordinates of the object point, P^O, $(x^{O_2}, y^{O_2}, z^{O_2})$ or $(x^{O_1}, y^{O_1}, z^{O_1})$ can be determined from its projections on the two image planes $\langle \pi_1 \rangle$ and $\langle \pi_2 \rangle$ [equations (1.3), (1.4) and (1.7)].

A word of caution. It has to be ensured while using the model to determine the 3−D co-ordinates of an object point, that the projections on the image planes are from the same point. This is known as the classical correspondence problem.
Now, let P_{π_1} be an image point on $\langle\pi_1\rangle$. It is actually, say, the projection of an object point P^O along the line l_{π_1} on $\langle\pi_1\rangle$. However, these facts are not enough to ascertain the exact location of the projection of the point P^O on $\langle\pi_2\rangle$; this is because P^O can lie anywhere on l_{π_1}. This, in other words, shows that, just by making use of geometrical constraints, the exact position of an object point on the image plane of the other sensor cannot be determined. However, use of the epipolar constraint [38], [39], [43], [42] can constrain the search area of locating the projection of P^O on $\langle\pi_2\rangle$. According to the epipolar constraint, the projection of P^O on $\langle\pi_2\rangle$ will be on the line of intersection of the plane formed by F_{π_1}, F_{π_2} and P_{π_1} with the plane $\langle\pi_2\rangle$. This means that if the projection of an object point, P^O, on $\langle\pi_1\rangle$ is known, then the epipolar constraint can be used to locate the projection of the same point on $\langle\pi_2\rangle$ by searching along a line in a particular direction.

The complexities associated with the solving of the classical correspondence problem can be reduced through the use of an active method, instead of a passive one [39]. In the case of active methods based on structured light projection, one of the two optical sensors is usually replaced with a structured light pattern projector. The problem then reduces to finding the elements of the structured pattern on the image plane of the sensor used to capture the pattern, for subsequent analysis of the pattern's deformations. It should be mentioned here that explicitly coding the structured pattern further reduces the problem of finding the elements of the pattern on the image plane of the sensor and helps in increasing the working volume of a typical depth sensing arrangement [39]. However, encoding the pattern also brings with it some problems which will be discussed later in this chapter.
For some significant work in the field of active depth sensing, from the nineteen-seventies to the mid nineteen-nineties, ranging from projection of a slit line scanning a screen, to grid pattern projection covering the entire field of view of the camera refer to the research reported in [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54]. Before discussing the various types of explicitly encoded structured light pattern

projection based range estimating arrangements, a general framework of a typical structured light projection based depth sensor is presented.

1.2.1 General principle of a structured light pattern projection based depth estimation arrangement

A global co-ordinate system, O_{π_1}, is fixed at the centre of the image plane, $\langle \pi_1 \rangle$, of the sensor used to grab the projected pattern and any deformations of it (Fig. 1.4). The co-ordinates of the focal point, F_{π_1}, of the sensor are given as $(0, 0, f_1)^T$ with respect to O_{π_1}. Now, since an arbitrary object point, P^O $\left[\equiv (x^O, y^O, z^O)\text{with respect to } O_{\pi_1}\right]$ will be projected on $\langle \pi_1 \rangle$ via the focal point, F_{π_1}, its projection P_{π_1} on $\langle \pi_1 \rangle$ and P^O can be related as follows:

$$P_{\pi_1} = F_{\pi_1} + p_1(P^O - F_{\pi_1}) \tag{1.8}$$

This can be expressed in matrix form as follows:

$$\begin{bmatrix} x_{\pi_1} \\ y_{\pi_1} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_1 \end{bmatrix} + p_1 \begin{bmatrix} x^O \\ y^O \\ z^O - f_1 \end{bmatrix} \tag{1.9}$$

where, in equation (1.9), $f_1 = \overline{O_{\pi_1} F_{\pi_1}}$. Another co-ordinate system, O_{π_2}, is fixed at the centre of the image plane, $\langle \pi_2 \rangle$, of the structured light pattern projector which is assumed to be working like a reverse camera [55]. Moreover, it is also assumed that O_{π_2} has the same orientation as that of O_{π_1}. The co-ordinates of O_{π_2} (centre of the co-ordinate system) can then be expressed in terms of the global co-ordinate system, O_{π_1}, as:

$$O_{\pi_2} = (x_2, y_2, z_2)^T \tag{1.10}$$

The focal point of the assumed reverse camera, F_{π_2}, can then be expressed in terms of the global co-ordinate system as:

$$F_{\pi_2} = (x_2, y_2, z_2 + f_2)^T \tag{1.11}$$

where, in equation (1.11), $f_2 = \overline{O_{\pi_2} F_{\pi_2}}$.

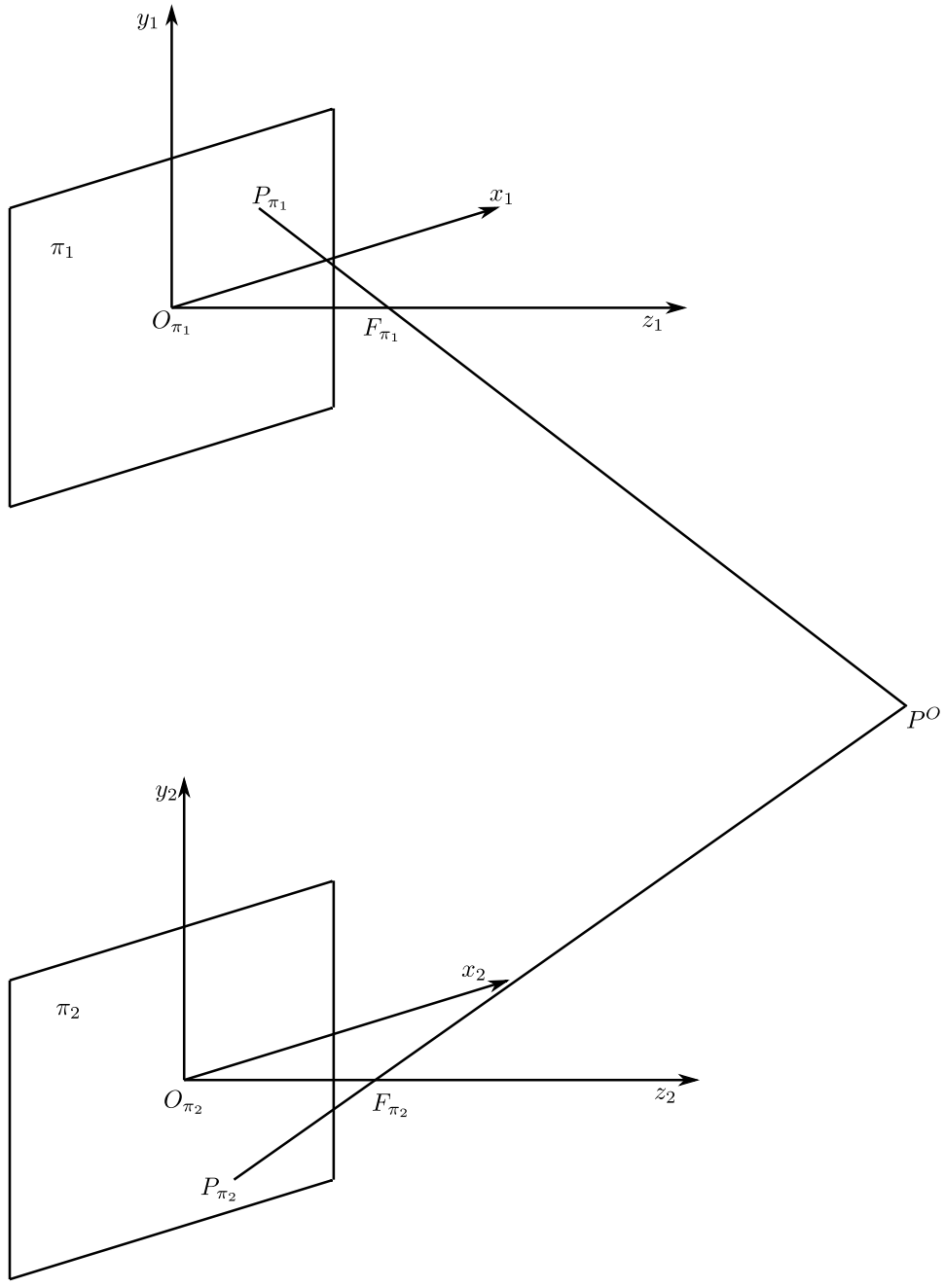


Figure 1.4: A general structured light based depth estimation arrangement; note that in the typical case shown, the co-ordinate system, O_{π_1}, attached to the image plane of the sensor and the plane, O_{π_2}, attached to the image plane of the projector have the same orientation.

Now, suppose the object point, P^O, is projected from the point P_{π_2} lying on the image plane $\langle\pi_2\rangle$ of the reverse camera set-up. The co-ordinates of P_{π_2} with respect to O_{π_2} is given as $(x_{\pi_2}, y_{\pi_2}, 0)^T$; with respect to the global co-ordinate system, O_{π_2}, this can be expressed as follows:

$$P_{\pi_2} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} x_{\pi_2} \\ y_{\pi_2} \\ 0 \end{bmatrix} = \begin{bmatrix} x_2 + x_{\pi_2} \\ y_2 + y_{\pi_2} \\ z_2 \end{bmatrix} \tag{1.12}$$

Since any point on $\langle\pi_2\rangle$ is projected via F_{π_2}, P^O and P_{π_2} can be related as follows:

$$P_{\pi_2} = F_{\pi_2} + p_2(P^O - F_{\pi_2}) \tag{1.13}$$

Equation (1.13) can be expressed in matrix form as follows:

$$\begin{bmatrix} x_2 + x_{\pi_2} \\ y_2 + y_{\pi_2} \\ z_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 + f_2 \end{bmatrix} + p_2 \begin{bmatrix} x^O - x_2 \\ y^O - y_2 \\ z^O - z_2 - f_2 \end{bmatrix} \tag{1.14}$$

From equation (1.9) it can be derived that:

$$p_1^{-1} = \frac{x^O}{x_{\pi_1}} = \frac{y^O}{y_{\pi_1}} = \frac{f_1 - z^O}{f_1} \tag{1.15}$$

This gives us:

$$x^O = (\frac{f_1 - z^O}{f_1})x_{\pi_1} \tag{1.16}$$

$$y^O = (\frac{f_1 - z^O}{f_1})y_{\pi_1} \tag{1.17}$$

Similarly, from equation (1.14) we get:

$$p_2^{-1} = \frac{x^O - x_2}{x_{\pi_2}} = \frac{y^O - y_2}{y_{\pi_2}} = \frac{f_2 + z_2 - z^O}{f_2} \tag{1.18}$$

From equation (1.18) we get:

$$x^O = x_2 + (\frac{f_2 + z_2 - z^O}{f_2})x_{\pi_2} \tag{1.19}$$

$$y^O = y_2 + (\frac{f_2 + z_2 - z^O}{f_2})x_{\pi_2} \tag{1.20}$$

Now, from equations (1.16) and (1.19) we get:

$$(\frac{f_1 - z^O}{f_1})x_{\pi_1} = x_2 + (\frac{f_2 + z_2 - z^O}{f_2})x_{\pi_2} \tag{1.21}$$

Therefore,

$$z^O = \left[\frac{x_2 + (1 + \frac{z_2}{f_2})x_{\pi_2} - x_{\pi_1}}{(\frac{x_{\pi_2}}{f_2} - \frac{x_{\pi_1}}{f_1})} \right]$$
$$= \frac{f_1 f_2}{f_1 x_{\pi_2} - f_2 x_{\pi_1}} \left[x_2 + x_{\pi_2} - x_{\pi_1} + \frac{z_2 x_{\pi_2}}{f_2} \right] \quad (1.22)$$

Similarly, equating equations (1.17) and (1.20) we get:

$$z^O = \frac{f_1 f_2}{f_1 y_{\pi_2} - f_2 y_{\pi_1}} \left[y_2 + y_{\pi_2} - y_{\pi_1} + \frac{z_2 y_{\pi_2}}{f_2} \right] \quad (1.23)$$

It becomes apparent from equations (1.22) and (1.23) that depth, z^O, of the object point, P^O, can be estimated given the co-ordinates of the image point on $\langle \pi_2 \rangle$ from which P^O is projected, P_{π_2}, and the co-ordinates of the projection of P^O on $\langle \pi_1 \rangle$, P_{π_1}. Of course, co-ordinates of the centre of O_{π_2} with respect to the global co-ordinate system, O_{π_1}, and the focal lengths, f_1 and f_2, have to be *a priori* determined through a calibration process.

Note also from equations (1.22) and (1.23) that determination of either the x- coordinates, x_{π_1} and x_{π_2}, or the y- coordinates, y_{π_1} and y_{π_2}, are enough to estimate the depth of the object point P^O. This redundancy can be exploited by making either the x- coordinates or the y- coordinates of the projected pattern carry more information (coded) to solve the, otherwise difficult, classical correspondence problem.

Also note that in the general framework described, it is assumed that the two co-ordinate systems, O_{π_1} and O_{π_2}, have the same orientation. This means that only 3 elements of the translational vector, $\mathbf{T}$, have to be *a priori* determined to map O_{π_2} with O_{π_1}. If this is not the case, then a thorough calibration process has to be undertaken to describe O_{π_2} in terms of the global co-ordinate system, O_{π_1}. The following subsection will show how the focal lengths, the translations and the orientations of the separate optical devices (co-ordinate systems associated with them) can be expressed in terms of the global co-ordinate system which is usually fixed on one of the devices used in the entire arrangement.

1.2.2 Calibration process

A relatively straightforward calibration scheme can be developed by locating several object points and their corresponding projections on an image plane. Using homo-

geneous co-ordinates [42], the entire perspective transformation can be described as:

$$\begin{bmatrix} f^w x_{\pi_1} \\ f^w y_{\pi_1} \\ f^w \end{bmatrix} = \begin{bmatrix} R_{11}^\Theta & R_{12}^\Theta & R_{13}^\Theta & R_{14}^\Theta \\ R_{21}^\Theta & R_{22}^\Theta & R_{23}^\Theta & R_{24}^\Theta \\ R_{31}^\Theta & R_{32}^\Theta & R_{33}^\Theta & R_{34}^\Theta \end{bmatrix} \begin{bmatrix} x^O \\ y^O \\ z^O \\ 1 \end{bmatrix} \tag{1.24}$$

where, in equation (1.24), f^w is a non-zero value and rotation, translation, scaling and perspective transformations are described by the matrix: $\begin{bmatrix} R_{11}^\Theta & R_{12}^\Theta & R_{13}^\Theta & R_{14}^\Theta \\ R_{21}^\Theta & R_{22}^\Theta & R_{23}^\Theta & R_{24}^\Theta \\ R_{31}^\Theta & R_{32}^\Theta & R_{33}^\Theta & R_{34}^\Theta \end{bmatrix}$

From equation (1.24) we get:

$$f^w x_{\pi_1} = R_{11}^\Theta x^O + R_{12}^\Theta y^O + R_{13}^\Theta z^O + R_{14}^\Theta \tag{1.25}$$

$$f^w y_{\pi_1} = R_{21}^\Theta x^O + R_{22}^\Theta y^O + R_{23}^\Theta z^O + R_{24}^\Theta \tag{1.26}$$

$$f^w = R_{31}^\Theta x^O + R_{32}^\Theta y^O + R_{33}^\Theta z^O + R_{34}^\Theta \tag{1.27}$$

From equations (1.25) and (1.27) we get:

$$R_{11}^\Theta x^O + R_{12}^\Theta y^O + R_{13}^\Theta z^O + R_{14}^\Theta - R_{31}^\Theta x^O x_{\pi_1} - R_{32}^\Theta y^O x_{\pi_1} - R_{33}^\Theta z^O x_{\pi_1} - R_{34}^\Theta x_{\pi_1} = 0 \tag{1.28}$$

Similarly from equations (1.26) and (1.27) we get:

$$R_{21}^\Theta x^O + R_{22}^\Theta y^O + R_{23}^\Theta z^O + R_{24}^\Theta - R_{31}^\Theta x^O y_{\pi_1} - R_{32}^\Theta y^O y_{\pi_1} - R_{33}^\Theta z^O y_{\pi_1} - R_{34}^\Theta y_{\pi_1} = 0 \tag{1.29}$$

To simplify the calibration process, it is usually assumed that $R_{34}^T = 1$ [39]. Equations (1.28) and (1.29) can now be re-written as:

$$R_{11}^\Theta x^O + R_{12}^\Theta y^O + R_{13}^\Theta z^O + R_{14}^\Theta - R_{31}^\Theta x^O x_{\pi_1} - R_{32}^\Theta y^O x_{\pi_1} - R_{33}^\Theta z^O x_{\pi_1} = x_{\pi_1} \tag{1.30}$$

$$R_{21}^\Theta x^O + R_{22}^\Theta y^O + R_{23}^\Theta z^O + R_{24}^\Theta - R_{31}^\Theta x^O y_{\pi_1} - R_{32}^\Theta y^O y_{\pi_1} - R_{33}^\Theta z^O y_{\pi_1} = y_{\pi_1} \tag{1.31}$$

A vector, $\mathbf{U}$, is now defined by lexicographically arranging the elements of the transformation matrix, excluding the term R_{34}^T:

$$\mathbf{U} = \left[R_{11}^\Theta R_{12}^\Theta R_{13}^\Theta R_{14}^\Theta R_{21}^\Theta R_{22}^\Theta R_{23}^\Theta R_{24}^\Theta R_{31}^\Theta R_{32}^\Theta R_{33}^\Theta\right]^T \tag{1.32}$$

Collecting the other terms (not the elements of the transformation matrix) from the left-hand side of equation (1.30), a vector $\mathbf{v_x}$ is formed:

$$\mathbf{v_x} = \begin{bmatrix} x^O \\ y^O \\ z^O \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -x^O x_{\pi_1} \\ -y^O y_{\pi_1} \\ -z^O x_{\pi_1} \\ -x_{\pi_1} \end{bmatrix}^T \tag{1.33}$$

Similarly, collecting the other terms (not the elements of the transformation matrix) from the left-hand side of equation (1.31), a vector $\mathbf{v_y}$ is formed:

$$\mathbf{v_y} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ x^O \\ y^O \\ z^O \\ 1 \\ -x^O y_{\pi_1} \\ -y^O y_{\pi_1} \\ -z^O y_{\pi_1} \\ -y_{\pi_1} \end{bmatrix}^T \tag{1.34}$$

The x - coordinate of the projection of P^O on $\langle \pi_1 \rangle$, x_{π_1}, is then given by the scalar product of $\mathbf{v_x}$ and $\mathbf{U}$:

$$\mathbf{v_x U} = x_{\pi_1} \tag{1.35}$$

Similarly the y - coordinate of the projection of P^O on $\langle \pi_1 \rangle$, y_{π_1}, is given by the scalar product of $\mathbf{v_y}$ and $\mathbf{U}$:

$$\mathbf{v_y U} = y_{\pi_1} \tag{1.36}$$

It becomes apparent from equations (1.35) and (1.36) that given the co-ordinates of an object point and its corresponding projection on an image plane, two elements of $\mathbf{U}$ can be determined. So, by locating 6 non-coplanar object points and their corresponding projections on an image plane, the vector $\mathbf{U}$ can be fully determined [39]. If the co-ordinates of the object points and their corresponding projections on the reverse camera image plane can also be linked up by repeating the described calibration procedure, then the co-ordinate system fixed on the image plane of the reverse camera can be described in terms of the global co-ordinate system. Finally, it needs mentioning that though only 6 non-coplanar objects and their corresponding projections are required to fully determine the vector $\mathbf{U}$, according to many authors at least 60 points are necessary for an fairly accurate calibration [39].

1.2.3 'There is no such thing as a free lunch!'

From the above discussions it becomes apparent that replacing the optical sensor in a stereo-vision system by a structured light pattern projector can reduce the complexity involved in solving the correspondence problem. In fact, solving the problem can be made even trivial by explicitly encoding the projected structured pattern. Think of a structured pattern of dots where each dot is formed by an emitted light ray of a specific frequency within the visible range (colour coded) and using the same number of sensors as the number of dots in the pattern, each tuned to image a specific coloured dot and analyse its corresponding deformation. Ideally, in such a case, the correspondence problem solving issue might not be even spared a thought considering its contextual triviality. However, if the practical issues are taken into account, it would not be hard to realise that encoding the projected pattern in this way does not diminish the difficulty of the problem by any amount. First of all, if we consider, say, a 32×32 dot pattern formed of distinct hue values, then $(32 \times 32) = 1024$ sensors, each tuned to a particular frequency, are required to image the entire pattern. Moreover, if colours are projected on a scene, then its use will

be limited to generating surface maps of neutral colour objects [39]. A non-neutral colour scene may change the colour of some of the projected spots resulting in region losses in the perceived pattern. This indicates the fact that colour coding the projected pattern may make the working of the depth sensing arrangement scene dependent. There are of course many ways, other than colour, of explicitly encoding the projected pattern. Unfortunately, though each of these methods attempts to reduce the complexities involved in solving the correspondence problem, it carries some practical disadvantages with it. A broad overview of the various coding techniques and some of the limitations associated with each category are now presented. There are various ways of classifying the various structured pattern encoding techniques. It can be based upon temporal dependence [39], nature of light projected [39] or the limit put by the coding technique on the working range of the depth sensing system.

Based upon temporal dependence a coding technique can be classified into:

A. Static: This type of coding can be used while observing motion-less scenes [39]. Multiple patterns are usually projected to load more information on either of the two orthogonal co-ordinates (non-axial) of the projected pattern. This helps in locating the projected pattern elements on an image plane. However, since projecting multiple patterns takes time, any motion in the scene may jeopardise the working of the system.

B. Dynamic: If there is motion in the scene, usually a single coded pattern is projected [39]. While dealing with dynamic scenes, it has to be kept in mind that de-codification of a complex pattern is a time consuming process; on the other hand, projection of a relatively simple pattern limits the working volume of the system.

Coding techniques can also be categorised based on the nature of the light projected:

A. Binary: Presence or absence of light is used to encode a projected pattern element [39]. Usually such bi-valued patterns limit the working-volume of the depth sensing arrangement.

B. Grey-level: If the projected pattern is intensity coded, then decoding the scene

requires the scene to be first illuminated with constant light intensity and steps to cancel any surface reflection effect.

C. Colour: The hue values used to encode the elements of the pattern should be considerably different so that the colour constancy property can be properly exploited [39]. Such patterns usually make the working of the arrangement very much scene dependent.

It should be mentioned that a particular coding technique, or a coded pattern, may put some limit on the surface depth that can be measured with a certain degree of accuracy by a given depth estimation arrangement. In other words, it limits the working range of the depth sensing set-up. For example, if a particular pattern is repeated periodically, then to avoid ambiguity the surface depth range should not cause any deformity in a pattern element which is more than half of the period length. However, it is a common practice to periodically repeat a pattern as it reduces the number of bits required to codify the pattern. On the other hand, if the pattern is not repeated then there is, in theory, no restriction on the working range of the system. However, this advantage comes with higher cost and technical requirements which may not be feasible at all from a practical point of view.
For some interesting coding techniques that have been used to alleviate the difficulties associated with the solving of the correspondence problem, readers are referred to [56], [57], [58], [59], [60], [61] , [62], [63], [64], [65], [66], [67], [39]. For a more recent account of developments of structured light projection based depth sensing the reader is referred to [68], [69].

As it is now perhaps clear, even though explicitly encoding the projected pattern helps in reducing the complexity of the correspondence problem, it brings with it additional requirements and limitations. Looking at all these practicalities, perhaps, it is true that 'there is no such thing as a free lunch!'.

Keeping all these factors in mind, in Chapter 5 an attempt has been made, while modeling a structured light projection based active range estimation system, to increase the working volume of the arrangement without explicitly encoding the projected pattern.

In Chapter 6 it is shown how a foreground scene can be segmented without generating explicit depth maps and solely by using the projected pattern disparity information. Two median filtering schemes have also been developed in this chapter to de-noise unassociated pixel cluster contaminated foreground object masks generated using the projected spot position disparity based segmentation method.

1.3 A brief overview of the chapters

Chapter 2: The computational model to classify pixels in a scene has been developed in this chapter. It has been demonstrated how the computational model, with the optional use of a proposed channel ratio test, can be used to segregate moving-shadow regions in indoor scenes illuminated by a fixed incandescent source of light.

Chapter 3: The combined approach involving the computational model working in tandem with a standard background subtraction process has been constructed in this chapter. In addition, various popular outlier detection strategies have been studied and implemented to assess their suitabilities in generating a threshold, automatically, that can be used to develop a binary mask from a difference frame, the outcome of the background subtraction process. Finally, the chapter investigates the performance of the combined approach and compares it with that of the pixel-wise applied computational model through the use of some standard performance evaluation metrics.

Chapter 4: In this chapter the full scope of the pixel-labeling capabilities of the developed computational model has been assessed by applying the model as the first stage of a two-stage process to detect abandoned baggage in public places.

Chapter 5: An active structured light based depth-estimating system has been modeled in this chapter; the working of the developed model has also been demonstrated.

Chapter 6: The chapter outlines how the task of foreground scene segmentation can be accomplished by solely sensing the disparity in a projected structured, and without generating explicit depth-maps. Two custom-built median filtering schemes have also been formulated in this chapter to de-noise the difference

frames containing the silhouettes of the foreground objects, generated by the developed depth cue based scene segmentation process.

Chapter 7: In the final chapter the entire thesis has been discussed, some conclusions drawn and future work outlined.

Chapter 2

COMPUTATIONAL MODEL TO DETECT AND REMOVE MOVING-SHADOWS IN VIDEO IMAGES

2.1 Introduction

Motion segmentation is a powerful method for segmenting objects of interest in many computer vision scenarios. However, the moving object may also cast a moving shadow (false-target) which may become incorrectly labelled as part of the foreground object. Hence shadow detection in video images plays an important role in the overall efficiency and robustness of a real-time surveillance system. A method that works in the RGB colour space with a channel ratio test is described in this chapter. The core method, based on a Lambertian hypothesis, has been tuned to suppress highlights. Extensive statistical studies have been conducted to facilitate understanding, and also remove the *ad hoc* nature of the selected thresholds to the extent that the method may be coarse tuned. The method has been applied on various indoor video sequences, and the results show that it can be satisfactorily used to mark or eliminate the strong portion of the foreground shaded region.

2.2 Chapter organisation

The chapter has been organized in the following way: Section 2.3 illustrates how a shadow is formed, and why the detection and removal of moving-shadows is considered a vital step in the development of a video-surveillance system. The computational model developed is described in Section 2.4. Section 2.5 reports the results of an extensive statistical analysis done on the threshold parameters so as to help the end-user coarse tune these. Section 2.6 introduces the topic of channel ratio test that can be effectively used in certain video sequences to reduce false segmentations. The overall method is documented in Section 2.7. Sections 2.8 and 2.9 elaborates on the results and the significance of the channel ratio test. Conclusions are finally drawn in Section 2.10.

2.3 Shadow formation

A shadow is formed when light from a source is intercepted by an opaque body in such a way that the other side of the body not facing the source is in darkness. Projection of this dark region on a surface behind the object is known as a shadow region. Fig. 2.1 illustrates the formation of shadow due to an extended source smaller than the object. The part of the shadow that is devoid of any light is called the umbra, and the region surrounding it, which is partially dark, is called the penumbra or the soft portion of the shadow [28]. From another perspective, shadows can be categorized as static shadows or moving shadows [70]. Static shadows are cast by static objects that usually form a part of the background; their elimination has never been judged as a crucial pre-processing step, as such shadows usually do not jeopardize the actual foreground object recognition process of surveillance systems. On the other hand, shadows cast by dynamic objects or by objects suddenly brought into a background scene are often misclassified as the actual foreground objects, leading to poor object segmentation and tracking [70]. Hence, a foreground shadow region elimination process has become an unavoidable pre-processing step for the development and implementation of a robust and reliable real-time video-surveillance system.

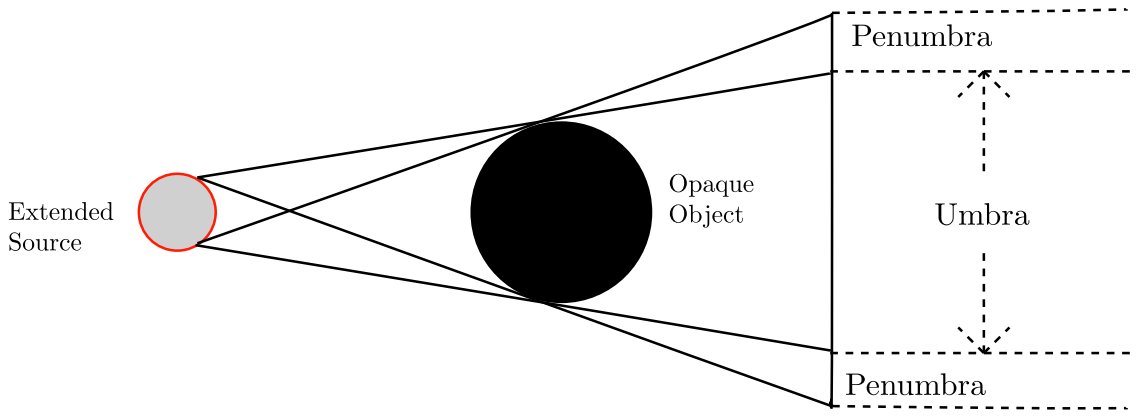


Figure 2.1: Shadow formation due to an opaque object larger than the extended source of light; the part of the shadow devoid of any light is called the umbra and the surrounding region is the penumbra.

2.4 Computational model

The colour model utilised is analogous to that developed by the authors of [21], and exploits the fact that a shadow can be considered as ' a semi-transparent region in the image, which retains a representation of the underlying surface pattern, texture or colour value' [21],[27]. The model estimates the brightness and the chromaticity distortion factors separately for each pixel of the current frame with respect to the corresponding pixel of the expected background frame. As illustrated in Fig. 2.2 , $\mathbf{E_i}$ is the expected colour vector of the i^{th} pixel in the RGB colour space, obtained after averaging N background frames, and $\mathbf{I_i}$ is the corresponding colour vector of the i^{th} pixel obtained from any one of the current frames. An estimate of the brightness distortion can be obtained by finding out the difference, $\mid \mathbf{E_i} \mid$, and the magnitude of the projection of $\mathbf{I_i}$ on $\mathbf{E_i}$. On the other hand, the chromaticity distortion factor for the i^{th} pixel can be estimated by determining the angle between $\mathbf{I_i}$ and $\mathbf{E_i}$.
The expected colour vector for the i^{th} pixel is given by:

$$\mathbf{E_i} = \bar{r_i}\hat{\mathbf{r}} + \bar{g_i}\hat{\mathbf{g}} + \bar{b_i}\hat{\mathbf{b}} \tag{2.1}$$

where, in equation (2.1), the bar indicates the mean pixel value measured over 100 frames and the hat indicates a unit vector along a specific colour axis. The current colour vector of the i^{th} pixel, obtained from any one of the current frames is given

as:

$$\mathbf{I_i} = r_i\hat{\mathbf{r}} + g_i\hat{\mathbf{g}} + b_i\hat{\mathbf{b}} \tag{2.2}$$

The projection of $\mathbf{I_i}$ on $\mathbf{E_i}$ is given by:

$$OP_i = \langle \mathbf{I_i}, \hat{\mathbf{e}}_\mathbf{i} \rangle \tag{2.3}$$

where, in equation (2.3), $\hat{\mathbf{e}}_\mathbf{i}$ is the unit vector in the direction of $\mathbf{E_i}$, and $\langle\rangle$ is the inner (dot) product of the two vectors. The difference, Ξ_i, between $\mid \mathbf{E_i} \mid$ and OP_i can be used as an estimate of the brightness distortion factor for the i^{th} pixel of the current frame:

$$\Xi_i = \mid \mathbf{E_i} \mid - OP_i \tag{2.4}$$

The chromaticity distortion factor for the i^{th} pixel is expressed as an angle, θ_i, between $\mathbf{E_i}$ and $\mathbf{I_i}$, is given as:

$$\theta_i = \arccos\left(\frac{\langle \mathbf{E_i}, \mathbf{I_i} \rangle}{\mid \mathbf{E_i} \mid\mid \mathbf{I_i} \mid}\right) \tag{2.5}$$

2.5 Statistical analysis of the observed data

Four different sequences were chosen, as shown in Fig. 2.3, to determine a threshold for the brightness distortion factor. For every sequence a region of interest (roi), covering the foreground shadow region, was chosen interactively. The brightness distortion suffered by each of the pixels in that region was determined using equation (2.4). Subsequent statistical analysis revealed that the data, $\{\Xi\}$, was negatively skewed (Fig. 2.4) and the histograms, typically having a single peak, are of the form shown in Fig. 2.5. It was noted that the chosen threshold, was roughly equal to the standard deviation of the data subtracted from its mean value. However, it should be mentioned that the typical form of the histogram clearly depicts that the data does not follow any single standard distribution; this was confirmed by the formal chi-squared goodness-of-fit tests [71].

The data was then redistributed to 100 bins of equal width. The class-marks for each of the bins were then determined. It was observed that the selected threshold

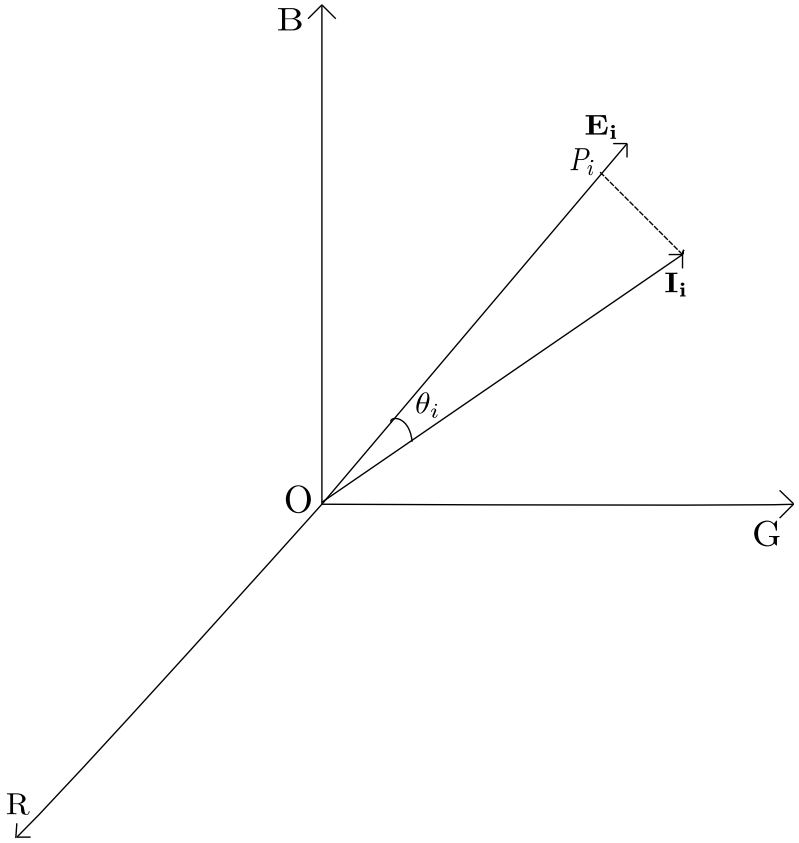
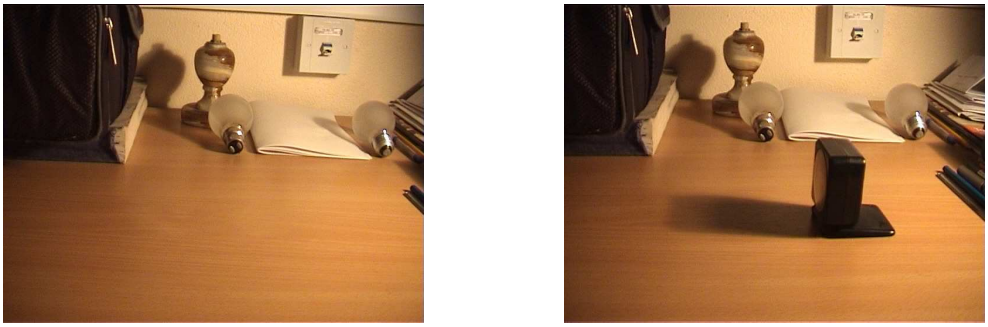
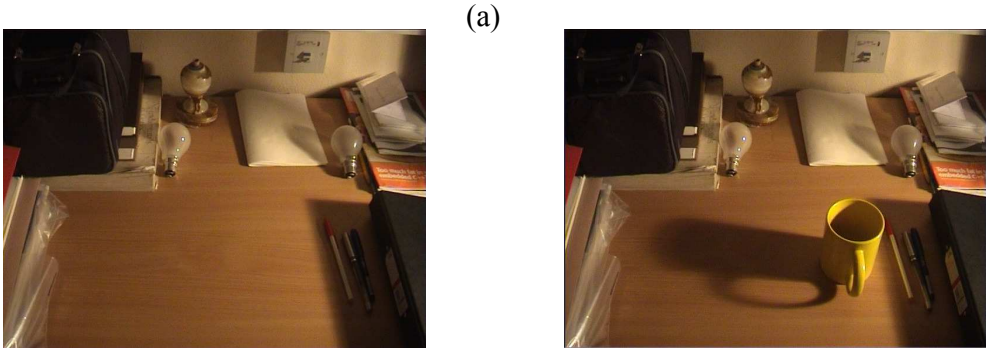


Figure 2.2: The computational model in the RGB colour space; $\mathbf{E_i}$ is the expected colour vector, $\mathbf{I_i}$ is the current colour vector, OPi is the projection of $\mathbf{I_i}$ on $\mathbf{E_i}$, and θ_i is the angle between $\mathbf{I_i}$ and $\mathbf{E_i}$.



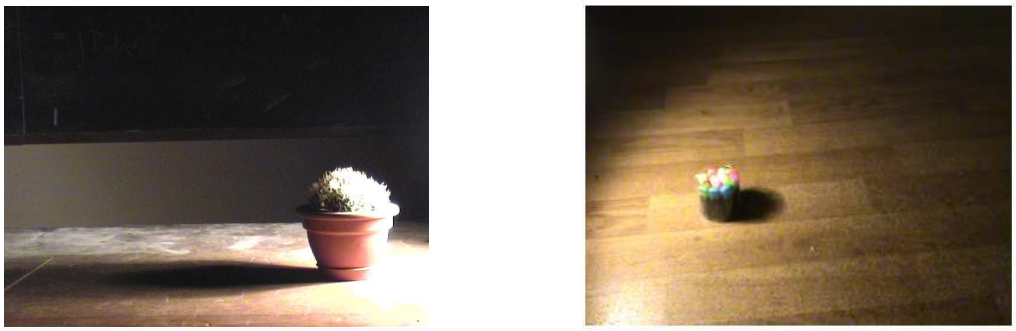
(i) (ii)

(a)



(i) (ii)

(b)



(c) (ii) (d) (ii)

Figure 2.3: For four examples (a), (b), (c) and (d): (i) expected background frame (only shown for (a) and (b) for compactness); (ii) one of the current frames.

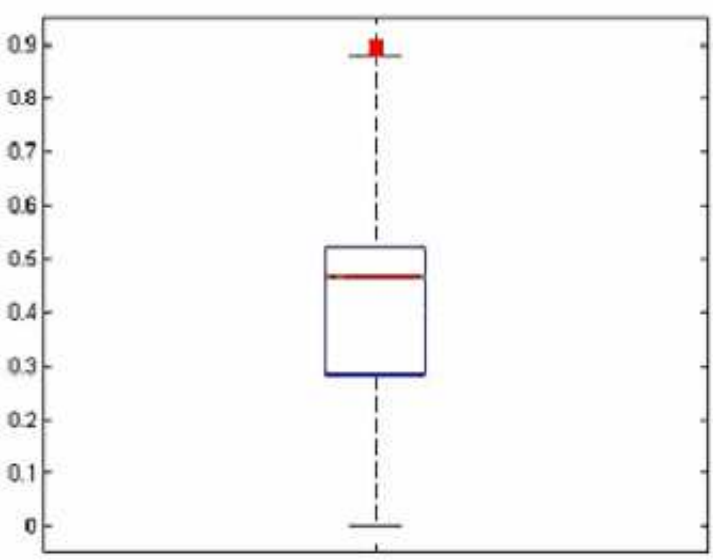


Figure 2.4: Boxplot for $\{\Xi\}$ depicts that the data is negatively skewed, as the difference between the median and the first quartile is more than the difference between the third quartile and the median; observations lying outside the inner fence are the suspected outliers.

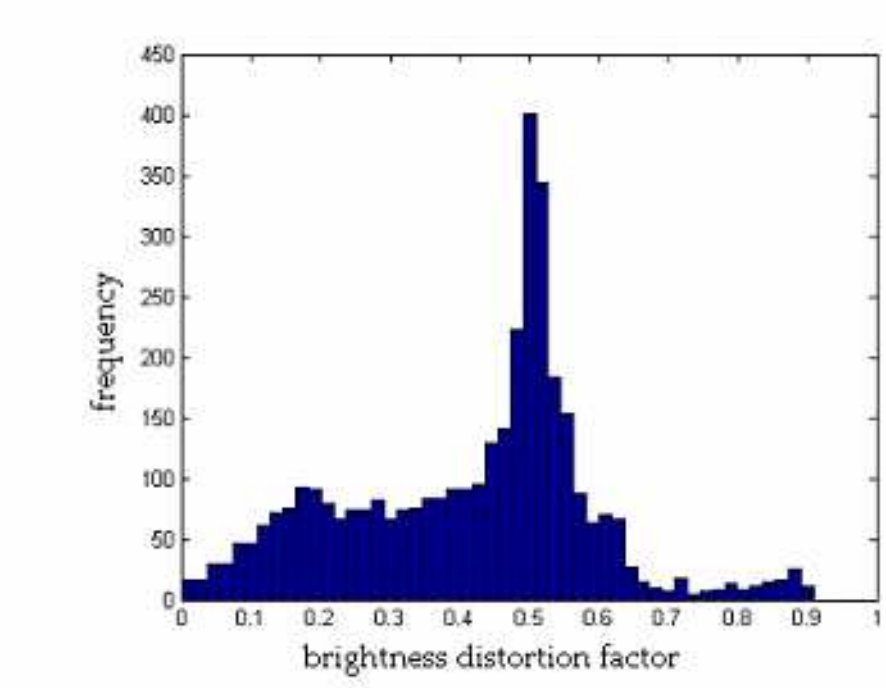


Figure 2.5: Typical histogram for the brightness distortion data, $\{\Xi\}$; the data being distributed among 50 bins of equal width.

nearly corresponded to the class-mark that segregates the data in a 24:76 ratio for the first sequence, 27:73 ratio for the second, 25:75 for the third, and 22:78 for the fourth. According to the scheme, one of the criteria of marking a given pixel as a shadow pixel is that the brightness distortion factor value for that pixel must be greater than the selected threshold. The analysis thus reflects the fact that at least 70% of the pixels in the shadow region suffer a brightness distortion which is above the chosen threshold.

It was observed that the threshold for the chromaticity distortion factor was roughly equal to the mean of the data added to the standard deviation of the same (data). Note a similar initial procedure was then adopted to analyse the chromaticity distortion factor, θ_i. Histograms were constructed based on the observed values of the parameter. A model comprising two normals with weights p_1 and p_2 was chosen to fit the constructed histogram (Fig. 2.6). The probability distribution function of the model is given as:

$$f(\theta_i) = \frac{p_1}{\sqrt{2\pi}\sigma_1} e^{\left(-\frac{(\theta_i - \mu_1)^2}{2\sigma_1^2}\right)} + \frac{p_2}{\sqrt{2\pi}\sigma_2} e^{\left(-\frac{(\theta_i - \mu_2)^2}{2\sigma_2^2}\right)} \quad \forall\ i \text{ in the roi} \qquad (2.6)$$

where, in equation (2.6), μ_1 and σ_1 are the mean and standard deviation of the normal distribution with weight p_1; μ_2 and σ_2 are those of the other normal distribution with weight p_2.

Maximum likelihood estimates (MLEs) of the five unknown parameters (weight: p_1, means: μ_1 and μ_2, and standard deviations: σ_1 and σ_2) were then determined using MATLAB [72]. A Kolmogorov-Smirnov (KS) test [73] was then carried out to test the null hypothesis that the underlying distribution has the form expressed in equation (2.6). It was found that the value of the test statistic was less than the critical value at a significance level of 0.05; hence, it can be concluded that there is no evidence against our proposed distribution at the 5% level.

It should be mentioned that the algorithm uses the assumption that the chromaticity distortion factor of the pixels in the shadow region will be less than a chosen threshold, τ_θ. Investigations were carried out based on the assumed mixture model to find out the probability of getting a θ_i value less than the chosen threshold in the shaded region. Thus, let θ_M be the measured value of θ_i, where θ_i represents any

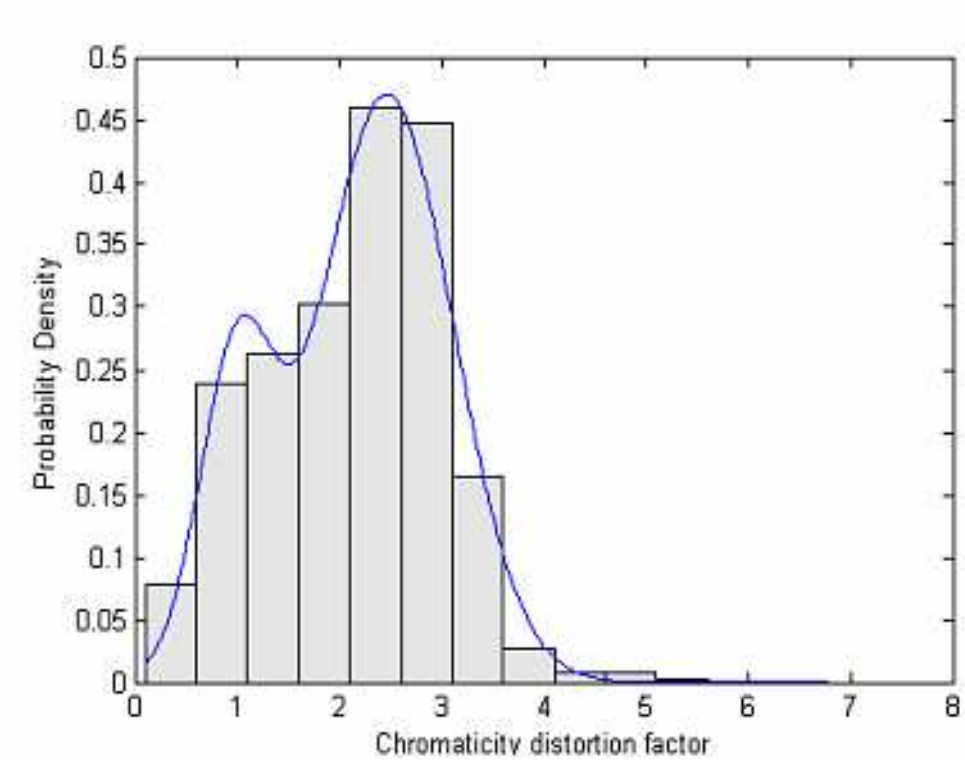


Figure 2.6: Superimposed on the scaled histogram is the curve corresponding to equation (6) with its MLE parameters.

random pixel in the chosen area; it is defined as follows:

$$\theta_M = \begin{cases} \theta_X, & \text{with probability } p_1 \\ \theta_Y, & \text{with probability } p_2 \end{cases} \tag{2.7}$$

where, in equation (2.7), θ_X and θ_Y are random variables with the following distributions:

$$\theta_X \sim N(\mu_1, \sigma_1) \tag{2.8}$$

$$\theta_Y \sim N(\mu_2, \sigma_2) \tag{2.9}$$

Now, the probability of getting a value of θ_M less than the chosen threshold is given as:

$$\begin{aligned} P(\theta_M < \tau_\theta) =& p_1 P(\theta_X < \tau_\theta) + p_2 P(\theta_Y < \tau_\theta) \\ =& p_1 P(\frac{\theta_X - \mu_1}{\sigma_1} < \frac{\tau_\theta - \mu_1}{\sigma_1}) + p_2 P(\frac{\theta_X - \mu_2}{\sigma_2} < \frac{\tau_\theta - \mu_2}{\sigma_2}) \\ =& p_1 P(z < \tau_\theta^{'}) + (1 - p_1) P(z < \tau_\theta^{''}) \end{aligned} \tag{2.10}$$

where, in equation (2.10), z is the standard normal variable, $\tau_\theta^{'} = (\frac{\tau_\theta - \mu_1}{\sigma_1})$ and $(\frac{\tau_\theta - \mu_2}{\sigma_2})$.

After computing the maximum likelihood estimates of the various parameters in equation (2.10) for each of the four sequences, it was revealed that there is approximately more than an 80% chance of getting a value of the chromaticity distortion factor for any pixel in the shadow region which is less than τ_θ.

Thus, to get the initial (starting) values of the threshold parameters the following 2-step process has to be followed:

1. Compute the mean and the standard deviation of the brightness distortion data, and take the threshold as the standard deviation subtracted from the mean.

To confirm (optional step):

Distribute the brightness distortion data into 100 *bins of equal width, and then determine the class-mark the divides the data into a* $30 : 70$ *ratio. Ascertain the class-mark is roughly equal to the chosen threshold value.*

2. Compute the mean and the standard deviation of the chromaticity distortion factor data, and take the threshold as the mean added to the standard deviation.

To confirm (optional step):

Fit a mixture model comprising two normals to the constructed histogram using the chromaticity distortion data, and then find out whether $P(\theta_M < \tau_\theta)$ *or not.*

2.6 Channel ratio test

It was observed that when the scene was illuminated by an indoor incandescent source the intensity of the red channel was more than that of the green or the blue channel for each individual pixel. Moreover, it was also observed that if the light source is blocked by an opaque object, resulting in the formation of a shadow on a non-coloured surface, then for more than 80% of the pixels in the shaded region:

$$\left(\frac{\Gamma_{bi}}{\Gamma_{Bi}}\right) > \left(\frac{\Gamma_{ri}}{\Gamma_{Ri}}\right) \tag{2.11}$$

where, in equation (2.11), Γ_{bi} and Γ_{ri} are the intensities of the blue and red channels, respectively, of the i^{th} pixel in the shaded region; Γ_{Bi} and Γ_{Ri} are the intensities of

the blue and red channels, respectively, of the i^{th} pixel of the expected background frame. This ratio test when adopted as an integral part of the developed algorithm to detect the actual shadow region on a non-coloured natural surface was found to reduce the false detection rate significantly and thus increase its efficiency, as will be demonstrated below. In this context, it is worth mentioning that shadows cast by objects on an outdoor neutral surface during day light tend to be more bluish in colour. This is because the only source of illumination on an outdoor shadow region is the sky. This observation has been utilized to mark the shadow pixels by Nadimi *et. al.* [70].

2.7 The proposed algorithm for shadow detection

N frames of the background are taken and averaged to get the expected background frame. This is done since, due to camera sensor noise, the RGB colour value of any given pixel, I_i, does not remain constant for all the N background frames. Any one of the current frames should then be considered and a pixel by pixel search undertaken to generate a shadow mask, Λ defined as:

$$\Lambda_i = \begin{cases} 1, & \text{if } (\Xi_i > \tau_b) \cap \left(\frac{\Gamma_{bi}}{\Gamma_{Bi}}\right) > \left(\frac{\Gamma_{ri}}{\Gamma_{Ri}}\right) \cap (\theta_i < \tau_\theta) \; \forall \; i, i \in [1, M \times N] \\ 0, & otherwise \end{cases} \tag{2.12}$$

where, in equation (2.12), $M \times N$ denotes the frame size.

It should be mentioned here that Ξ_i is to be calculated as equation (2.4) and a positive result ensured so as to suppress highlights; the algorithm thus works well for near-Lambertian surfaces. Morphological filters (erosion, dilation, and area) have to be deployed as the final step to remove noise, and to close the gaps in between the detected shadow regions. Note that henceforth we have used the phrase 'standard cleaning process' to mean application of the combination of morphological filters.

2.8 Results

Four sequences (a), (b), (c) and (d) were taken as shown in Fig. 2.3. The results after applying each of the criteria of marking a shadow pixel are shown for sequence

(a). As can be seen from Fig. 2.7 (i) — (iii), application of each of the criteria helps in reduction of noise. The final result after applying a standard cleaning process is shown in Fig. 2.7 (iv). For the other sequences, (b), (c) and (d), the final results after applying the standard cleaning process are demonstrated in Fig. 2.7 (v) — (vii). A careful observation of the results reveals the fact that the applied method mainly marks the umbra portion of the shadow.

The results also infer that to cover the entire shadow area threshold values have to be relaxed. This objective could be achieved through the deployment of a prior foreground object segmentation process to generate a binary-mask; the generated mask could then be used to search moving shadow pixels with relaxed threshold values as the constrained mask region would arrest the rise of the false detection rate [74]. However, such binary-mask based shadow search process would eventually reduce the speed of the entire process [75].

2.9 Significance of the channel ratio test

Investigations were also carried out as a part of the work to evaluate the significance of the channel ratio test. In this regard, it should be borne in mind that the efficiency of a given technique can be quantitatively evaluated, firstly, by assessing its effectiveness as a good shadow point detector i.e. by determining what is the probability of misclassifying an actual foreground shadow point as a non shadow point; also, secondly, by determining its discrimination potential i.e. by finding out the probability of classifying a non-foreground shadow point as an actual foreground shadow point. Such assessment in our case was done by applying the developed scheme twice for each of the captured video sequences; once with the test part included and then with this excluded. The two separate results obtained for each of the sequences were then examined and compared. Two metrics have been defined to quantify the performance. One of the two metrics is termed the shadow detection rate, ξ , and is defined as follows:

$$\xi = \frac{\eta_{dfs}}{\eta_{fs}} \tag{2.13}$$

where, in equation (2.13), η_{dfs} is the total number of actual foreground shadow pixels detected by the scheme and η_{fs} is the total number of actual foreground

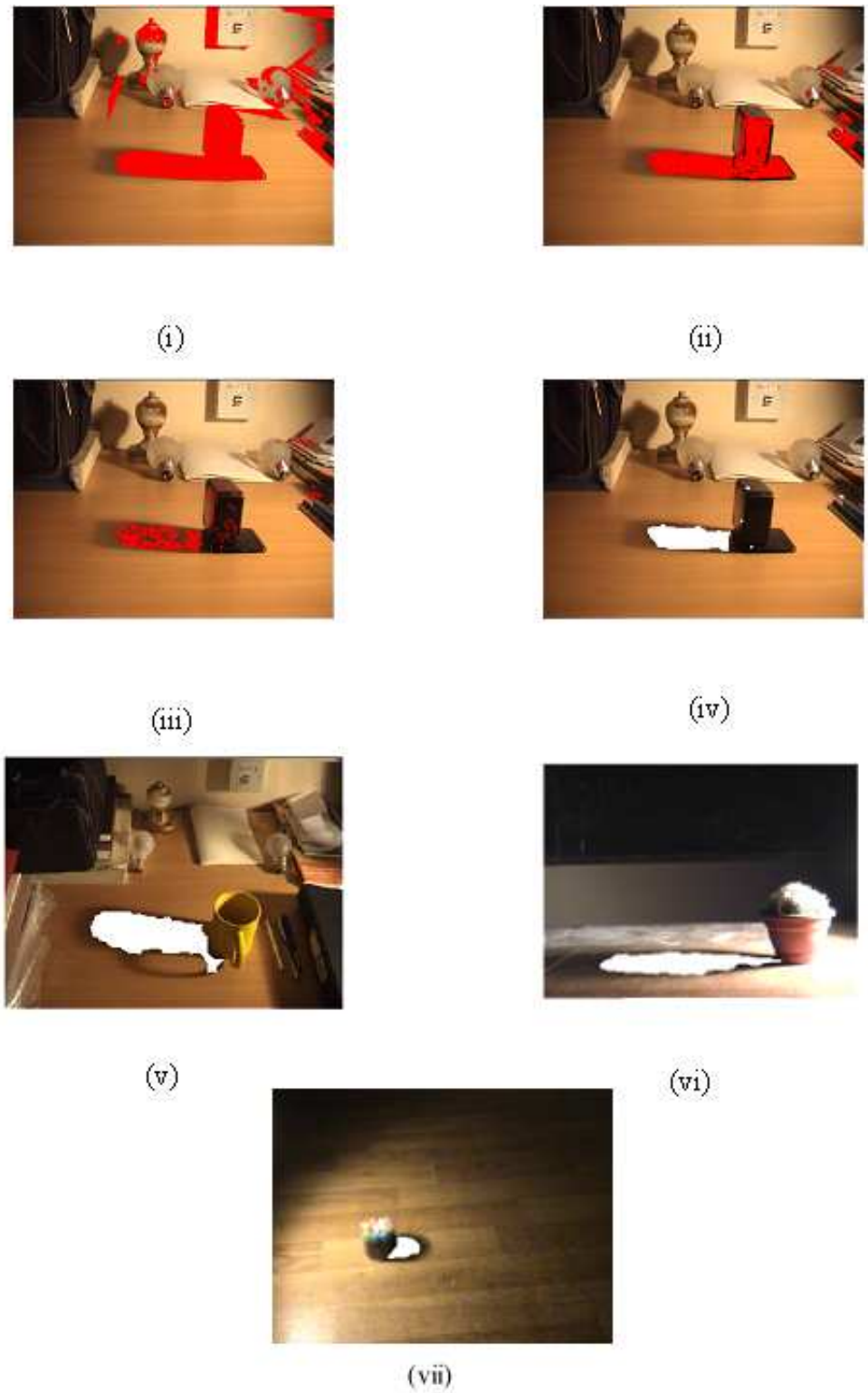


Figure 2.7: (i) (iii) Results after applying each of the marking criteria (brightness distortion test, channel ratio test and chromaticity distortion test) on video sequence (a); (iv) after cleaning the result obtained in (iii); (v) result after applying the overall method and the cleaning process on video sequence (b); (vi) result after applying the overall method and the cleaning process on video sequence (c); (vii) result after applying the overall method and the cleaning process on video sequence (d).

shadow pixels. From the definition it is evident that the range of ξ is between 0 and 1. In the ideal case the total number of detected foreground shadow pixels should be equal to the total number of actual foreground shadow pixels and, then, $\xi = 1$. In the practical case, the more efficient the shadow detection scheme, the closer to 1 will be the value attained by ξ.

Another metric is the false detection rate, λ, which quantifies the discriminatory potential of the utilised method. It is defined as follows:

$$\lambda = \frac{\eta_{ds} - \eta_{dfs}}{\eta_{ds}} \tag{2.14}$$

where, in equation (2.14), η_{ds} is the total number of pixels detected as shadow points by the scheme. The range of λ also lies between 0 and 1. Here, in the ideal case, the difference $\eta_{ds} - \eta_{dfs}$ should be 0 and then λ will be equal to 0. However, in reality a value close to 0 indicates the efficient performance of the method. The values of the metrics were found for the four different video sequences, considering only the umbra portion of the shadow in each case and the results have been tabulated in Table 1. As has been evinced in Table 1, a substantial decrease in the false detection rate can be achieved through the use of the ratio-test. Note that we also observed a reduction in the shadow detection rate when the channel ratio test was incorporated in the overall test (Fig. 2.8).

2.10 Summary

In this chapter a method for detecting foreground shadow pixels from video sequences of indoor scenes, illuminated in each case by a single fixed incandescent source, is described. Relevant results of the statistical analysis undertaken have been reported so as to remove the *ad hoc* nature of the thresholds to be selected, to a certain extent. The approach can be utilised for real-time scene activity tracking if the expected background frame and the thresholds can be obtained as parts of an offline process.

It should be noted that the spectrum of the light coming from an indoor incandescent source has more power towards the red end of the spectrum; hence, on a neutral surface, it was found that the intensity of the red channel is more than that of the green and the blue channel. It can be inferred from the channel ratio test

Table 2.1: Values of ξ and λ for the three different video sequences.

Video sequences	Scheme where the channel ratio test has been employed has been marked E; otherwise NE.	ξ (expressed as %)	λ (expressed as %)
(a)	E	83.79	7.84
(b)	E	86.23	6.89
(c)	E	89.57	4.54
(d)	E	81.01	3.17
(a)	NE	91.87	37.49
(b)	NE	92.24	22.54
(c)	NE	96.74	24.85
(d)	NE	93.02	17.47

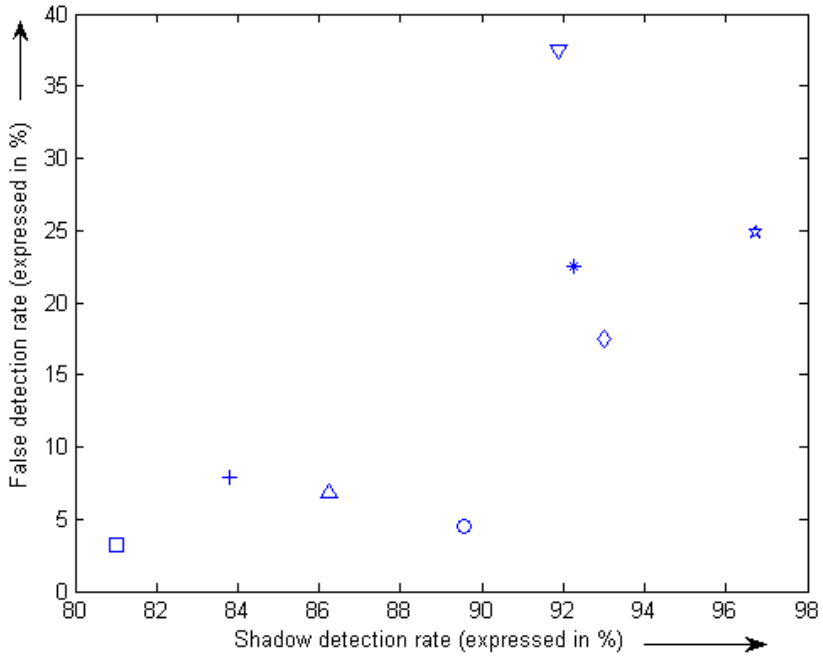


Figure 2.8: False detection rates plotted against shadow detection rates for the various video sequences concerned; plus: video sequence (a), channel ratio test employed; up-triangle: video sequence (b), channel ratio test employed; circle: video sequence (c), channel ratio test employed; square: video sequence (d), channel ratio test employed; down-triangle: video sequence (a), channel ratio test not employed; 8-pointed star: video sequence (b), channel ratio test not employed; 5-pointed star: video sequence (c), channel ratio test not employed; diamond: video sequence (d), channel ratio test not employed.

that the red channel suffers significant reduction compared to its expected intensity value with respect to that of the blue channel in shadow regions, cast by objects blocking an indoor incandescent source, on a non-coloured natural surface.

The developed model requires a single current frame to compare it with the expected background in order to mark the shadow pixels. Thus it can also be utilised to remove foreground shadow regions in digital still images, provided an image of the background without the actual foreground object is available.

The final results obtained indicated that the algorithm mainly marks the strong portion (umbra) of the shadow. Region growing techniques to encompass the soft portion of the shadow and also segmentation methods in order to expedite the overall process will be described in the next chapter. Also, note that that the model can also be used to segment actual foreground objects from the background. This will be explained in Chapter 4 where it will be explained how the developed computational model can be modified to be used as part of a two-stage approach to detect abandoned objects in public places.

Chapter 3

SHADOW ELIMINATION AFTER MOVING REGION SEGMENTATION BASED ON DIFFERENT THRESHOLD SELECTION STRATEGIES

3.1 Introduction

A popular approach for detecting moving object regions in video sequences is the application of the background subtraction technique. According to this technique the background (reference) image is subtracted from the current image frame and the moving parts are detected by the selection of a suitable threshold. In this chapter a method is presented to discriminate the moving pixels of the generated difference images from the relatively stationary pixels through the use of three different threshold selection strategies, namely: (i) '3σ edit' rule, (ii) rule utilizing the Hampel identifier, and (iii) rule based on an *ad hoc* selection of threshold. Further, after segmentation, a method of classification, based on the computational model described in the previous chapter, has been applied to segregate the moving shadow region from the actual moving object. The speed-up achieved through the use of the three aforementioned techniques on the core moving shadow search process,

compared to that where no such process has been applied, is reported. The final outcomes of applying the shadow detection technique after segmenting using each of the threshold selection strategies, one at a time, on some indoor video sequences are demonstrated and comparison of the methods made.
In the last section of the chapter a thorough comparison is drawn between the modified moving shadow elimination method with the pixel-wise shaded region determination method discussed in the previous chapter. The two methods have been compared on the basis of some standard metrics: effectiveness as a shadow detector; discrimination potential; and processing time; note all these performance assessing parameters can be quantitatively evaluated.

3.2 Chapter organisation

The entire chapter is organised in the following way: a preliminary idea about outliers and some popular outlier detection strategies are described in the next section. The following section describes how such detection strategies have been utilized to generate the binary mask from the difference image, and how further classification is made. In Section 3.5 results obtained through the use of a prior segmentation method are demonstrated and compared with the results where no such method has been applied. Note that the assessment of the modified method and the original one presented in this section are mainly based on subjective analysis. Only the processing times required by both the methods to do the core shadow search have been evaluated and compared quantitatively.
Finally, in Sections 3.6 and 3.7 the performance of the overall modified moving shadow elimination method has been gauged and compared with the unmodified method using some standard quantitative metrics. Conclusions are drawn in Section 3.8.

3.3 Outliers and outlier-detection strategies

Outliers are data points, $\{\pi_k\}$, in a data set, $\Pi = \{d_i\}$, ($\{\pi_k\} \subset \Pi$), that do not comply with our expectations based on the bulk of the data [32]. Various detection strategies have been thought of to detect such aberrant data points for

data preprocessing or filtering. Such strategies include visual inspection of data values of the set, fitting of models of desired form to the data and then examining the residuals or using deletion diagnostic approaches, etc [32]. However, efforts made towards the detection of outliers using the above mentioned strategies may prove to be futile due to the reasons described in [32] and, even if possible, are not applicable to meet our objective, as not only is the data set we are dealing with typically large but also due to the fact that the overall process has severe time constraints. In this respect, it should be mentioned that the visual inspection method of outlier detection cannot even be considered, as the overall process of moving region detection and classification has to be inherently automatic.

Fortunately, there are some other popular approaches for outlier detection. These well known and frequently used strategies depend on two estimates: (1) an estimate of a nominal reference value for the data set, and (2) a scatter estimate of the data. Based on these estimators, outliers can be detected based on the following criterion:

$$\mid d_i - \bar{d} \mid > \tau\gamma \Rightarrow d_i \in \{\pi_k\},\ \forall d_i \in \Pi \tag{3.1}$$

where, in equation (3.1), $\bar{d}$ is the nominal reference value of the dataset, τ is the threshold parameter, and γ the scatter estimate.

3.3.1 The '3σ edit' rule

The '3σ edit' rule considers the mean of the data values of the data set as the nominal reference value and the corresponding standard deviation as an estimate of the scatter:

$$\bar{d} = \bar{\Pi} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{3.2}$$

where, in equation (3.2), N is the total number of observations in the dataset.

$$\gamma = \sqrt{\left[\frac{1}{N-1}\sum_{i=1}^{N}(d_i - \bar{d})^2\right]} \tag{3.3}$$

It should be noted that if the distribution is assumed to be approximately normal, then the probability of getting a data value greater than three times the standard

deviation of the data, added to the mean, is around 0.3% [32]. However, the technique suffers from the fact that both the mean and the standard deviation of the data are very much outlier sensitive [32]. Moreover, the strategy heavily depends on the assumption that the underlying distribution is approximately Gaussian.

3.3.2 Strategy based on Hampel Identifier

This strategy capitalises on the fact that the outlier sensitive mean and standard deviation estimates are replaced by the outlier resistant median (breakpoint value of 50%) and median absolute deviation from the median (MAD) scale estimates, respectively. The median of a data sequence is obtained as follows [71]:

1. The observations are ranked according to their magnitude.
2. If N is odd, the median is taken as the value of the $\left[\frac{(N+1)}{2}\right]^{th}$ ranked observation; otherwise if N is even, the median is taken as the mean of the $\left[\frac{N}{2}\right]^{th}$ and $\left[\frac{N}{2}+1\right]^{th}$ ranked observations.

The MAD scale estimate, MAD_{se}, is defined as:

$$\gamma = MAD_{se} = \text{median}\{\mid d_i - d_{median} \mid\} \tag{3.4}$$

where, in equation (3.4), d_{median} is the median value of the dataset Π. Note that the MAD scale estimate is often scaled by multiplying it by a factor of 1.4826. This is done to make the MAD scale estimate an unbiased estimate of the standard deviation for normally distributed data [32].

$$\widehat{MAD_{se}} = 1.4826 \times \text{median}\{\mid d_i - d_{median} \mid\} \tag{3.5}$$

where, in equation (3.5), $\widehat{MAD_{se}}$ is the scaled MAD scale estimate.

The strategy, although quite often very effective in practice [32], is stymied by the fact that if more than 50% of the observations are of the same value, then the scale estimate is equal to 0, i.e. every data value greater than the median would then be considered as an outlier.

In this context, it should be noted that the mean can also be replaced by the median and the standard deviation by the interquartile deviation, giving rise to the so called standard boxplot outlier detection strategy [76].

3.4 Application of the technique

A number of background (period of relative inactivity [27]) frames, Ω, are taken, and the expected reference frame is calculated by employing an averaging process. This is done as the RGB colour value of any given pixel, Λ_i, does not remain constant for all the Ω background frames due to sensor noise. The difference image, DiffImage, is then generated by subtracting the expected background frame from the current frame. After that a binary mask is generated either by applying the '3σ edit' rule, the strategy utilising the Hampel Identifier, or by selecting a threshold on an *ad hoc* basis. A search algorithm, based on the computational model without the channel ratio test, is then deployed based on the binary mask to mark or eliminate the foreground shadow region of the current frame. A foreground pixel, Λ_{fi} is marked as a foreground shadow pixel, Λ_{fsi}, based on the following criterion:

$$\Lambda_{fi} \Rightarrow \Lambda_{fsi} \text{ if } \left[\left(\Xi_i > \tau_b^{'}\right) \cap \left(\theta_i < \tau_\theta^{'}\right)\right] \forall \Lambda_i == 1 \text{ in the binary mask} \tag{3.6}$$

where, in equation (3.6), $\tau_b^{'}$ and $\tau_\theta^{'}$ are the chosen (relaxed) brightness distortion and chromaticity distortion thresholds, respectively.

Note that Ξ_i is calculated using equation (2.4) so as to suppress highlights. Morphological filters are then applied to remove noise and to close the gaps in-between the detected region.

It should also be mentioned that the foreground shadow pixels have also been determined by applying a similar search criterion to all the pixels of the current frame; this is done to determine the speed-up achieved over the entire frame search process by deploying any one of the threshold selection strategies.

3.5 Observations on results

Two video sequences (a) and (b) are taken as shown in Fig. 3.1. The binary masks generated by applying the '3σ edit' rule on the DiffImages of the 2 sequences are shown in Fig. 3.2 (i) — (ii). Fig. 3.2 (iii) — (iv) demonstrate the resultant binary mask obtained after deploying the threshold strategy based on the utilisation of the Hampel Identifier, and Fig. 3.2 (v) — (vi) show the binary masks for the two sequences generated by a low threshold value, selected on an *ad hoc* basis.

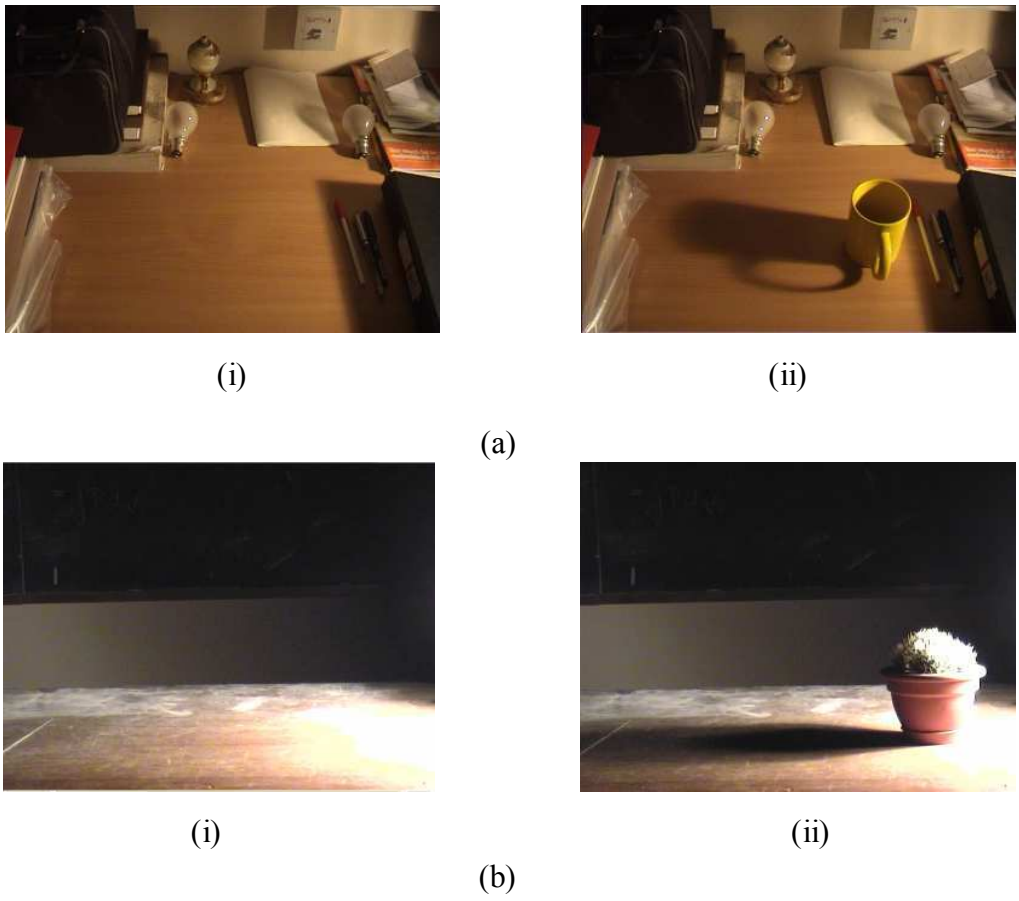


Figure 3.1: For the two indoor video sequences (a) and (b): (i) shows the expected background frame, and (ii) one of the object frames.

Fig. 3.3 (i) — (vi) reveals the final outcome after applying the foreground shadow search algorithm and cleaning process on all those points with a value of '1' in the binary mask. Fig. 3.4 (i) — (ii) demonstrates the results after applying a pixel-by-pixel foreground shadow search algorithm for the three sequences.

The normalized bargraph (Fig. 3.5) shows the time taken by the core search process, applied after the generation of the binary masks for the two sequences, with respect to the time taken where no such prior segmentation method was applied.

3.6 Performance metrics to evaluate the methods quantitatively

Investigations have also been carried out as a part of the work to compare the modified method with the original one. In this regard, it should be borne in mind

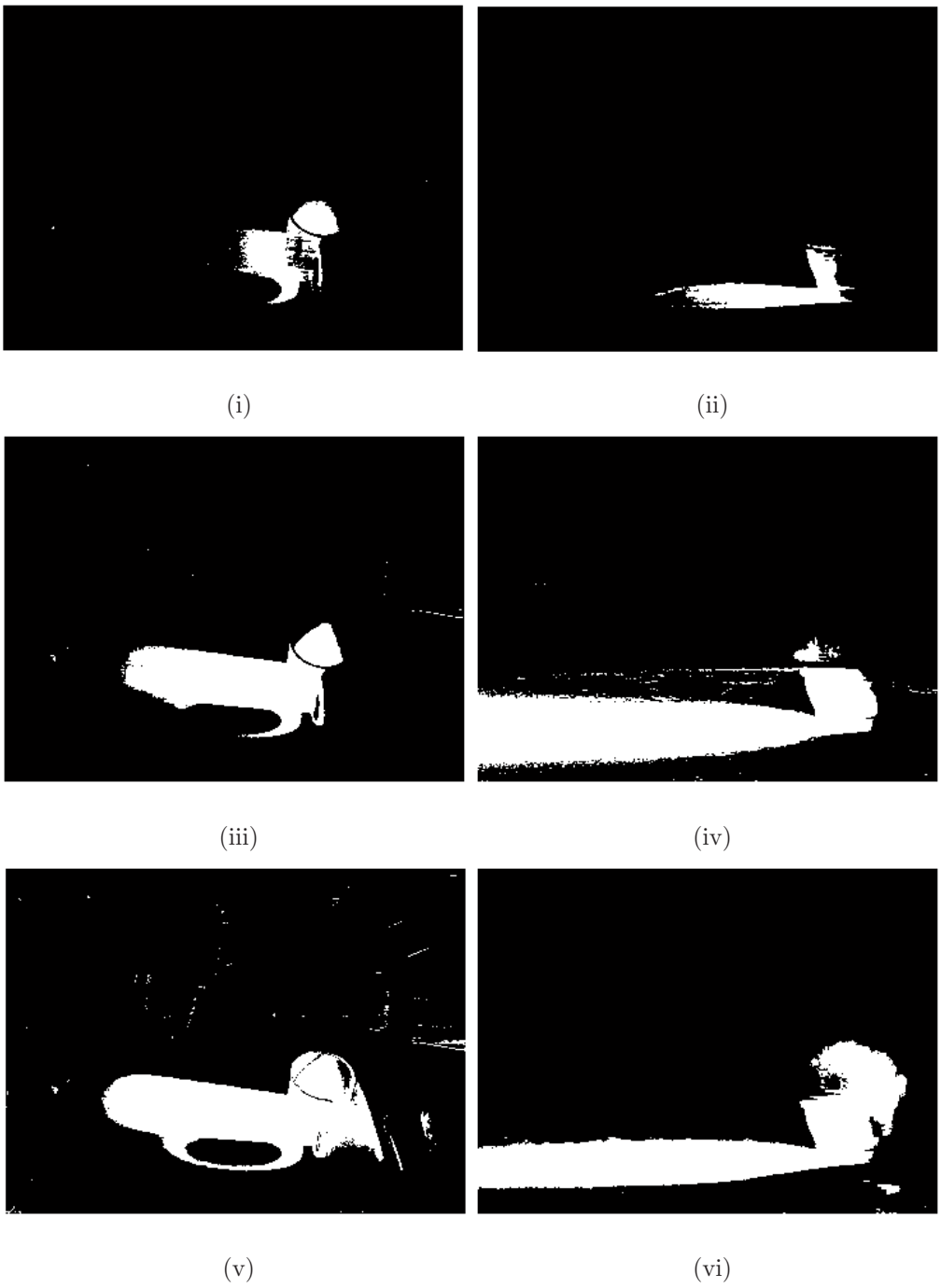
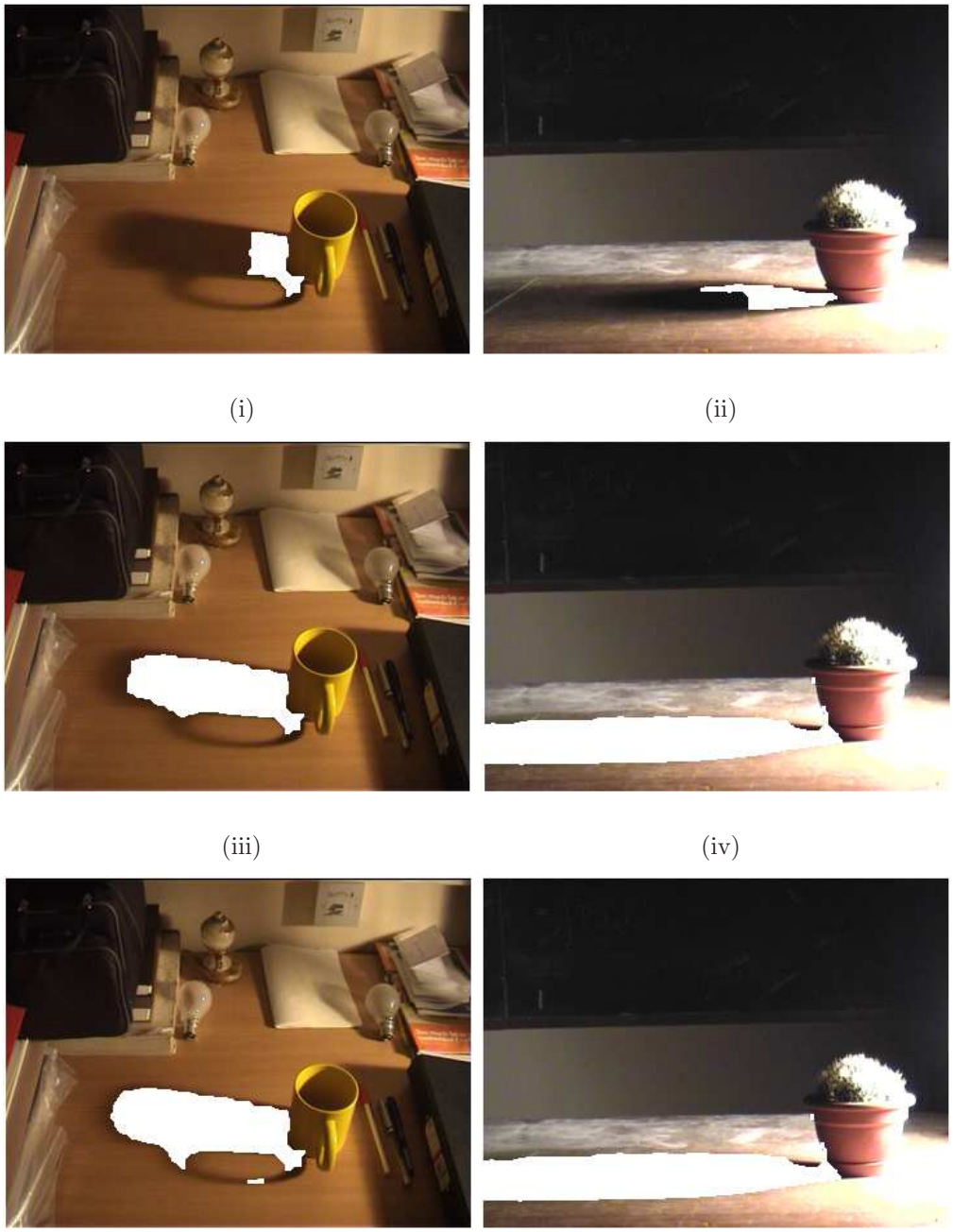


Figure 3.2: (i)—(ii) Binary masks for the video sequences (a) and (b) generated using the '3σ-edit' rule; (iii)—(iv) binary masks generated using the rule utilizing the Hampel Identifier; (v)—(vi) binary masks generated using a (low) threshold selected on an *ad hoc* basis.



(i) (ii)

(iii) (iv)

(v) (vi)

Figure 3.3: (i)—(ii) Results (marked white) after applying the shadow detection technique and cleaning process based on the binary mask generated using the '3σ-edit' rule; (iii)—(iv) results (marked white) after applying the shadow detection technique and cleaning process based on the binary mask generated using the rule utilizing the Hampel Identifier; (v)—(vi) results (marked white) after applying the shadow detection technique and cleaning process based on the binary mask generated using a (low) threshold selected on an *ad hoc* basis.

(i) (ii)

Figure 3.4: (i)—(ii) Results (marked white) where the cleaning process was applied after deploying a pixel-by-pixel shadow search algorithm.

that the efficiency of a given method can be quantitatively evaluated, as has been explained in the previous chapter, by assessing its effectiveness as a good shadow point detector i.e. by determining what is the probability of misclassifying an actual foreground shadow point as a non-shadow point; also, secondly, by determining its discrimination potential i.e. by finding out the probability of classifying a non-foreground shadow point as an actual foreground shadow point [22]. Values for both the parameters have been determined for all the video sequences for the two methods using metrics similar to those proposed in [77]. One of the two metrics is termed the shadow detection rate, ξ, and is defined as follows:

$$\xi = \frac{\eta_{dfs}}{\eta_{fs}} \tag{3.7}$$

From the definition it is evident that the range of ξ is between 0 and 1. In the ideal case the total number of detected foreground shadow pixels should be equal to the total number of actual foreground shadow pixels and, then, $\xi = 1$. In the practical case, the more efficient the shadow detection scheme, the closer to 1 will be the value attained by ξ. Another metric is the false detection rate, λ, which quantifies the discriminatory potential of the utilised method. It is defined as follows:

$$\lambda = \frac{(\eta_{ds} - \eta_{dfs})}{\eta_{ds}} \tag{3.8}$$

where, in equation (3.8), η_{ds} is the total number of pixels detected as shadow points by the scheme.

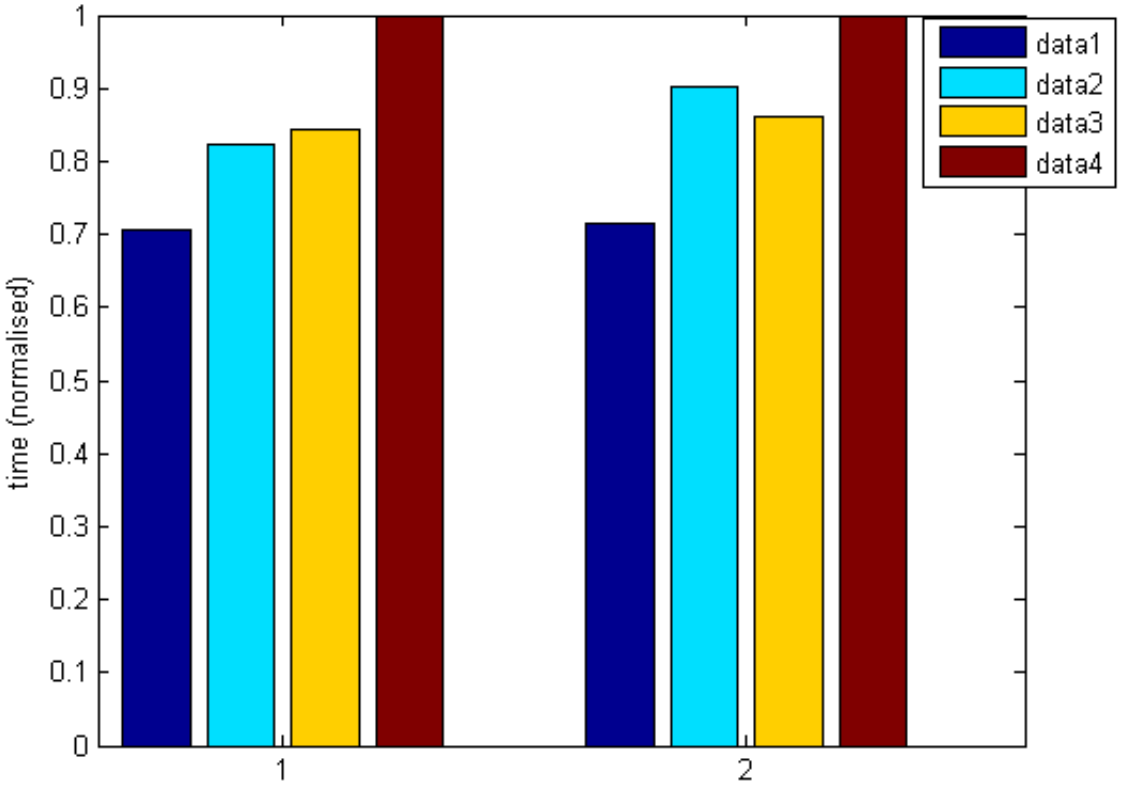


Figure 3.5: Normalised bars corresponding to data 1 ('3σ-edit' rule), data 2 (rule utilizing the Hampel Identifier) and data 3 (rule based on an *ad hoc* selection of threshold) depicts the time taken by the core shadow search process deployed after the generation of the binary masks with respect to that where no prior segmentation method was applied (data 4); the first group of bars (1) corresponds to video sequence (a) and the second group (2) corresponds to video sequence (b).

The range of λ also lies between 0 and 1. Here, in the ideal case, the difference $(\eta_{ds} - \eta_{dfs})$ should be 0 and then λ will be equal to 0. However, in reality a value close to 0 indicates the efficient performance of the method. The values of the metrics were found for the four different video sequences, considering both the strong and soft portions of the shadow in each case, and the results have been tabulated in Table 1.

Note that the metrics have also been described in Section 2.9 of the previous chapter; there the metrics have been used to gauge the significance of the ratio test that can be used as an integral part of the developed computational model to reduce the false detection rate.

The total times taken by the two methods (Table 2) to mark the moving shaded region have also been determined; τ_{pws} denotes the total time taken by the pixel-wise shadow search method, and τ_{bms} the total time taken by the modified method. For the modified method, the time taken by the segmentation/(binary-mask generation) process, τ_{seg}, and the time taken for the core shadow search process, τ_{csd}, have been determined separately. Note that:

$$\tau_{bms} = \tau_{seg} + \tau_{csd} \tag{3.9}$$

3.7 Results

Four sequences are chosen to analyse the performance of the modified method discussed in this chapter as compared to that of the pixel-wise method described in the previous chapter. The sequences are shown in Fig. 3.6 and Fig. 3.7. Note that out of the four sequences, only one (Fig. 3.6 (b)) has been used before. Inclusion of new sequences adds to the diversity of sequences on which the two methods have been tested.

Fig. 3.8 (i)—(iv) shows the final results after background retrieval based on the detected moving shaded region using a method that employs the '3σ edit' rule; it becomes quite evident that the method completely fails for all the sequences. This is because the binary-mask generated using the rule only partly covered the dynamic

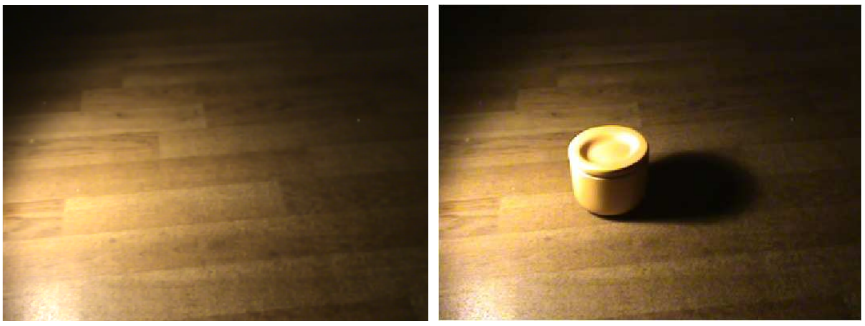
Table 1: Values of the two performance evaluation metrics: shadow detection rate (ξ) and false detection rate (λ) for the four video sequences after application of the pixel-wise moving shadow search methods, and the binary-mask based methods.

Video Sequences	Method1: Pixel-wise Moving Shadow Search Process		Method 2: Binary-mask Based Moving Shadow Search Processes			
	ξ	λ	Method where the binary-mask was generated though the use of a threshold chosen on an *ad hoc* basis		Method where the binary-mask was generated through the use of the Hampel Identifier	
			ξ	λ	ξ	λ
(a)	0.77	0.06	0.90	0.08	0.96	0.13
(b)	0.78	0.15	0.91	0.20	0.93	0.21
(c)	0.75	0.09	0.89	0.08	0.91	0.12
(d)	0.79	0.22	0.86	0.25	0.88	0.26

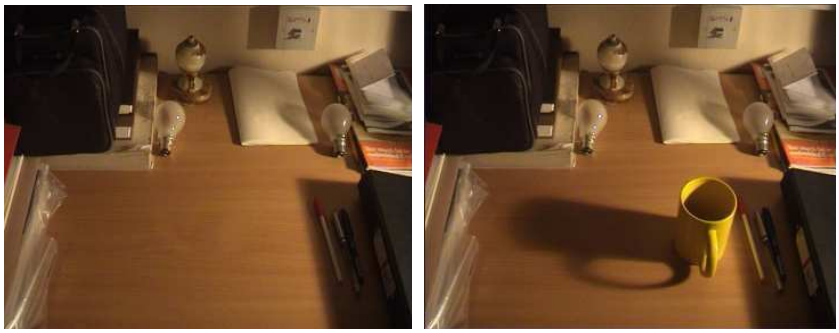
Table 2: The relative times taken by the segmentation/(binary-mask) generation processes, and the core moving shadow search processes with respect to the total times taken by the binary-mask based methods; also, included in the table are the relative times taken by the binary-mask based methods with respect to the total times taken by the pixel-wise shadow search method.

Video Sequences	Times taken in sec when the codes were executed in MATLAB τ_{pws} (s)	Relative times taken by the methods, and the sub-methods					
		Method where the binary-mask was generated though the use of a threshold chosen on an *ad hoc* basis			Method where the binary-mask was generated through the use of the Hampel Identifier		
		$f\tau 1$	$f\tau 2$	$f\tau 3$	$f\tau 1$	$f\tau 2$	$f\tau 3$
(a)	0.30	0.56	0.44	2.13	0.61	0.39	2.16
(b)	0.39	0.52	0.48	1.79	0.55	0.45	1.83
(c)	0.31	0.55	0.45	2.15	0.58	0.42	2.21
(d)	0.37	0.51	0.49	1.88	0.54	0.46	1.68

Note in Table 2, $f\tau 1 = \frac{\tau_{seg}}{\tau_{bms}}$, $f\tau 2 = \frac{\tau_{csd}}{\tau_{bms}}$, and $f\tau 3 = \frac{\tau_{bms}}{\tau_{pws}} = \frac{\tau_{seg} + \tau_{csd}}{\tau_{pws}}$

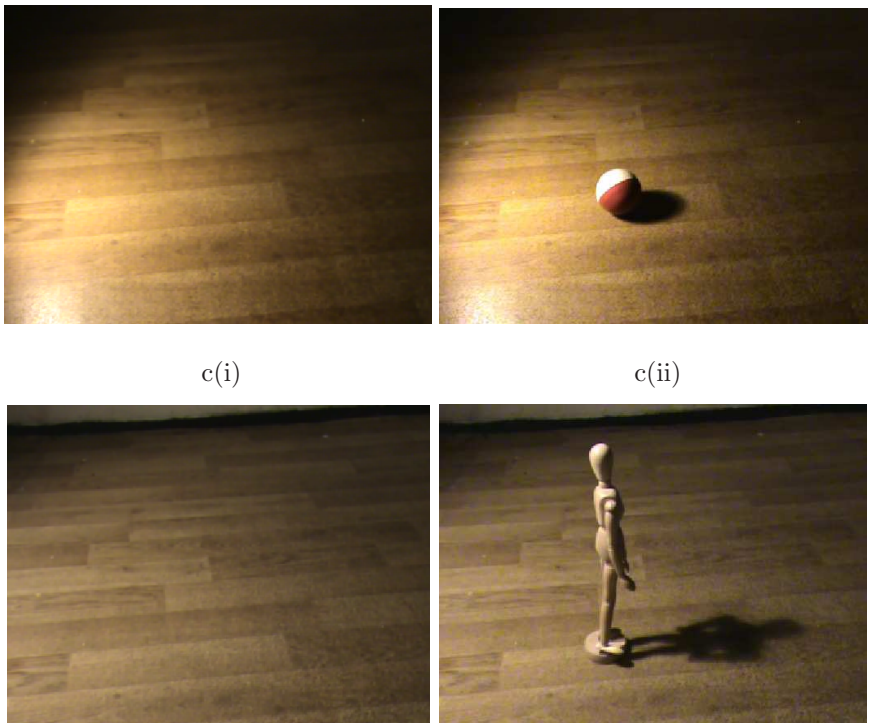


a(i) a(ii)



b(iii) b(iv)

Figure 3.6: For the sample indoor video sequences (a) Jar, (b) Cup: (i) shows the expected background frame, and (ii) one of the object frames.



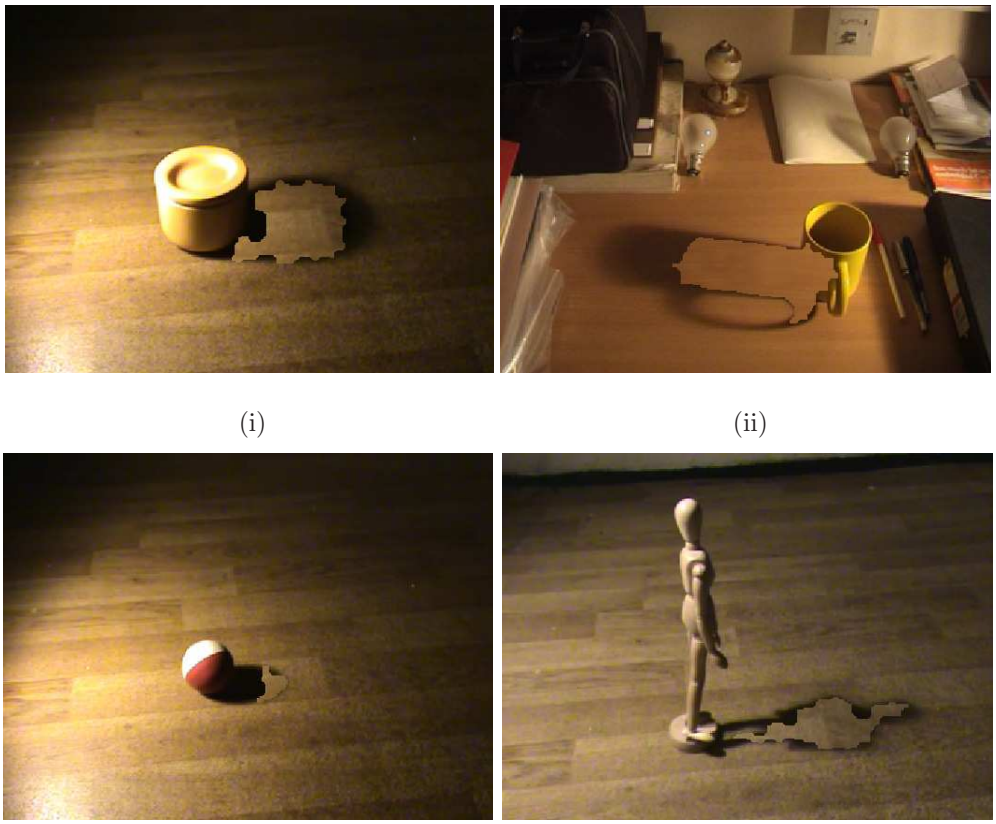
c(i) c(ii)

d(iii) d(iv)

Figure 3.7: For the sample indoor video sequences (c) Ball and (d) Mannequin: (i) shows the expected background frame, and (ii) one of the object frames.

region in the scene. In turn, this suggests that the underlying data distribution is not Gaussian.

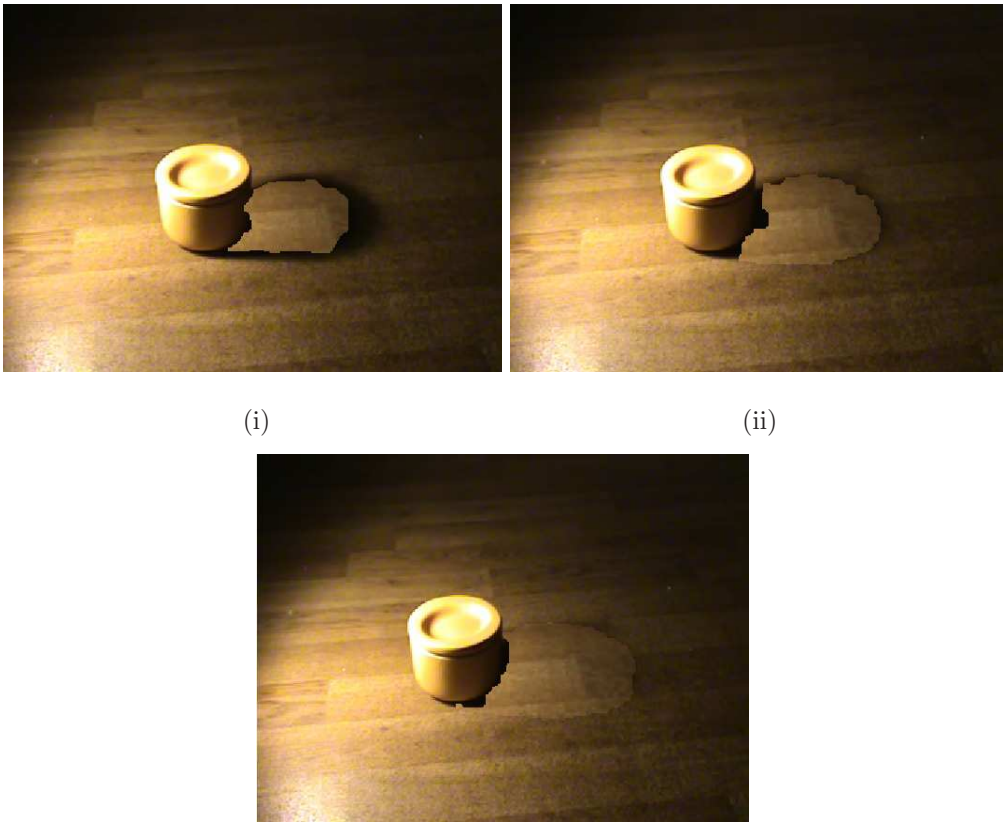
Fig. 3.9 — Fig. 3.12 depict the outcome after background retrieval based on the detected moving shaded region mask; the moving shaded region masks are obtained through the application of the pixel-wise shadow search method, and the other binary-mask based ones. Note the binary-masks are developed through the use of thresholds either chosen using the Hampel Identifier rule, or on an *ad hoc* basis. It becomes quite clear that the deployment of the binary mask based methods would help us to cover both the strong and soft portions of the moving shadows unlike the pixel-wise method which through the use of strict thresholds is able to encompass only the strong portions.



(i) (ii)

(iii) (iv)

Figure 3.8: Results after background retrieval based on the detected shaded region in video sequences: (a)—(d); the shaded regions were detected through the use of the binary-mask (generated through the deployment of the '3σ edit' rule) based shadow search method.

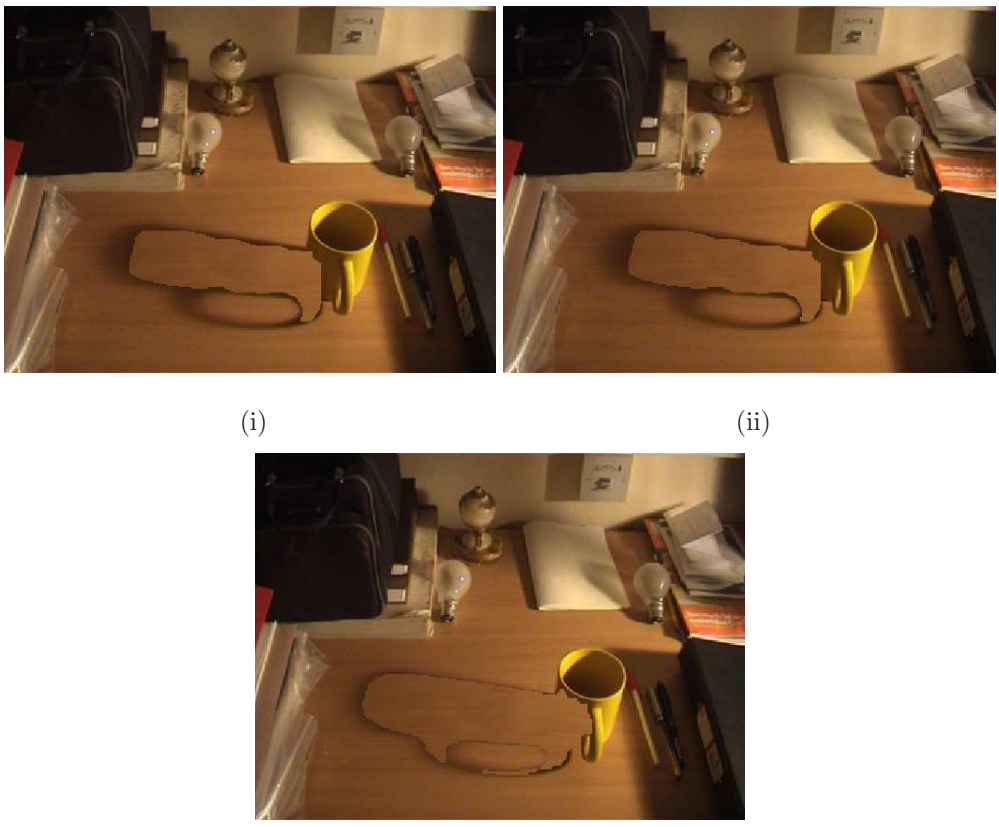


(i) (ii)

(iii)

Figure 3.9: Results after background retrieval based on the detected shaded region mask in video (a) using (i) pixel-wise moving shadow search process; (ii) binary-mask (generated through the use of a threshold chosen on an *ad hoc* basis) based moving shadow search process; (iii) binary-mask (generated through the use of the Hampel Identifier) based moving shadow search process.

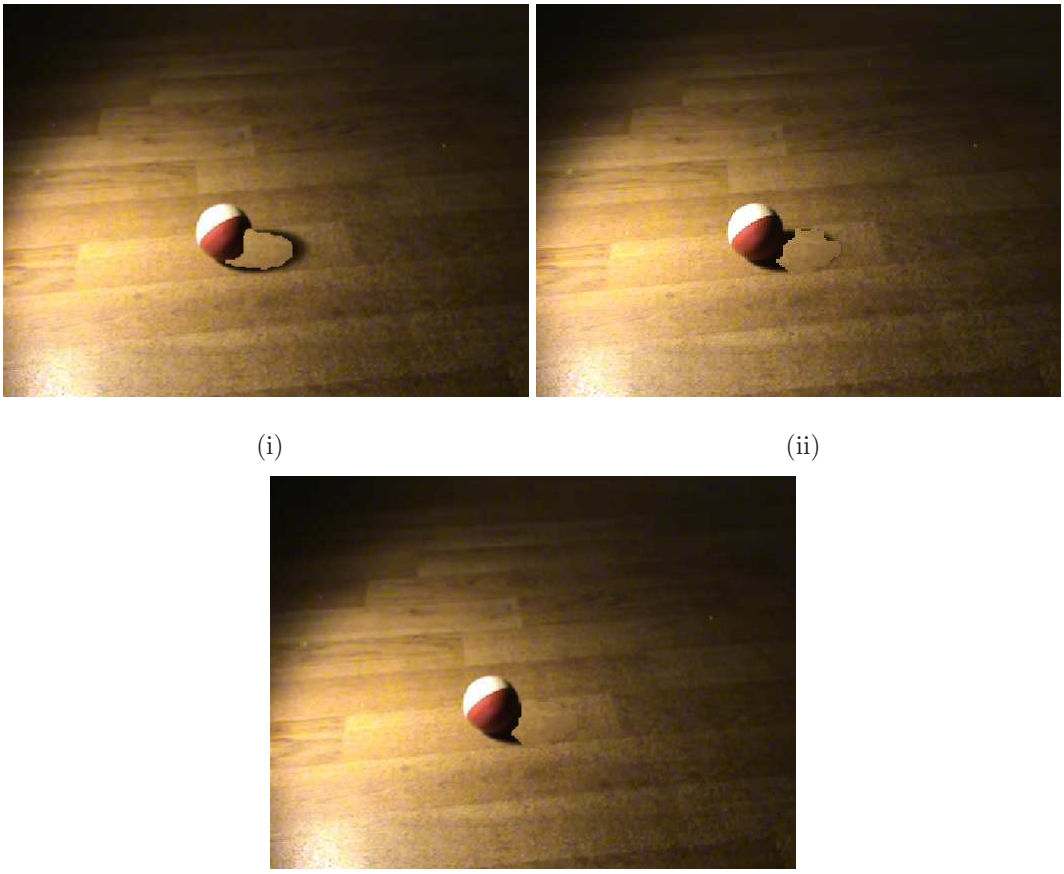
Table 1 lists the efficiency of each of the applied methods [shadow detection rate (ξ)] and the discrimination potential [false detection rate (λ)] obtained by the application of the methods. The occasional increment of the false detection rate in cases where the binary-mask based methods have been deployed occurs due to the fact that the methods use relaxed threshold values to cover the entire shadow area. The relative times taken by the binary-mask based methods with respect to the pixel-wise moving shadow search methods have been listed in Table 2. Table 2 also shows the relative (foreground object segmentation)/(binary-mask generation) times, and the core shadow search process times with respect to the total times, in the examples of the modified methods. It becomes quite obvious that the binary-mask generation



(i) (ii)

(iii)

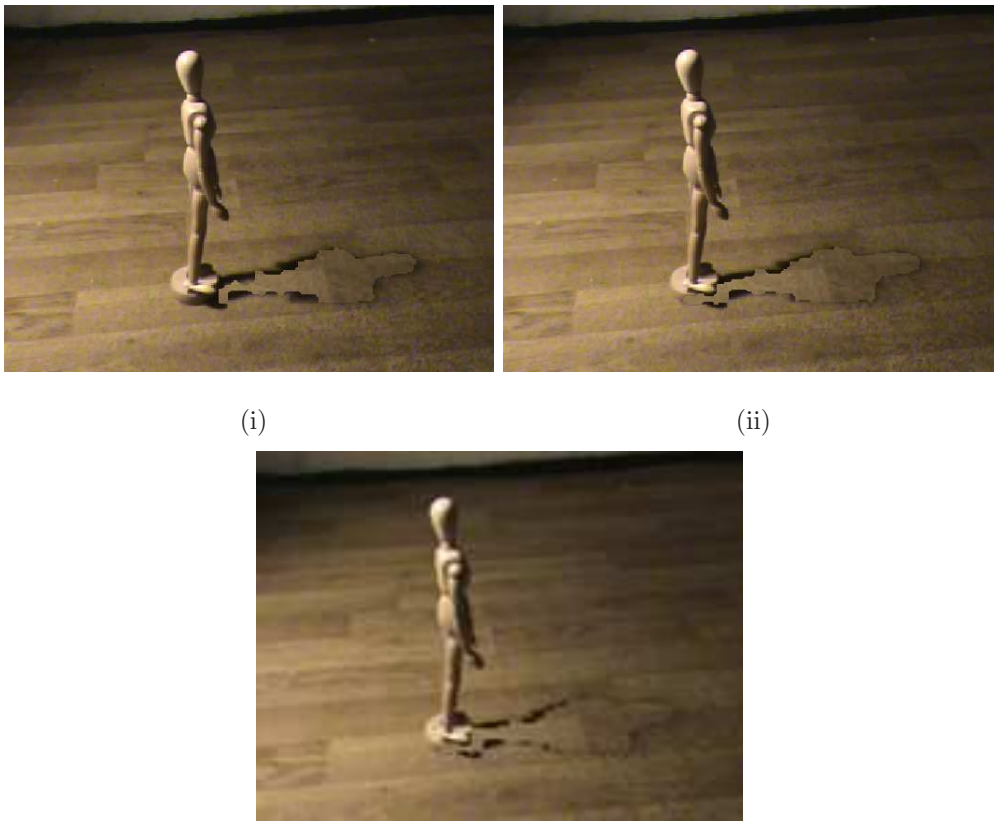
Figure 3.10: Results after background retrieval based on the detected shaded region mask in video (b) using (i) pixel-wise moving shadow search process; (ii) binary-mask (generated through the use of a threshold chosen on an *ad hoc* basis) based moving shadow search process; (iii) binary-mask (generated through the use of the Hampel Identifier) based moving shadow search process.



(i) (ii)

(iii)

Figure 3.11: Results after background retrieval based on the detected shaded region mask in video (c) using (i) pixel-wise moving shadow search process; (ii) binary-mask (generated through the use of a threshold chosen on an *ad hoc* basis) based moving shadow search process; (iii) binary-mask (generated through the use of the Hampel Identifier) based moving shadow search process.



(i) (ii)

(iii)

Figure 3.12: Results after background retrieval based on the detected shaded region mask in video (d) using (i) pixel-wise moving shadow search process; (ii) binary-mask (generated through the use of a threshold chosen on an *ad hoc* basis) based moving shadow search process; (iii) binary-mask (generated through the use of the Hampel Identifier) based moving shadow search process.

takes more than 50% of the overall time and makes such methods slower compared to the pixel-wise moving shadow region search methods. Note that the binary-mask based methods outperform the pixel-wise moving shadow search methods efficiency-wise. Moreover, it should be noted that though the modified methods take more time as compared to the pixel-wise search methods they could still be applied for real-time applications. The Hampel Identifier based method shows good results efficiency-wise and should be used to make the overall process inherently automatic.

3.8 Summary

In this chapter it has been described how the computational model, through the use of relaxed threshold values, can mark both the strong and soft portions of moving shadows in video images without loosing control on the false detection rate. It has also been noted that the outlier detection strategy employing the Hampel Identifier can be efficiently used to generate the binary masks from difference image frames. From the results documented it is clear that though the binary-mask based shadow search processes take less time compared to the pixel-wise shadow search process to do the core shadow search, the former methods take more time than the latter one to generate the final moving shadow masks (Table 2). This is because of the fact that a prior foreground object segmentation method together with the execution of a threshold detection and decision making strategy, in case of the binary-mask based methods, consume a considerable amount of time to generate the moving region masks. How the computational method can be modified to pick up only the real targets, neglecting the moving shadows, is described in the next chapter and a description of an application of the modified method is presented.

Chapter 4

A TWO-STAGE APPROACH TO DETECT ABANDONED BAGGAGE IN PUBLIC PLACES

4.1 Introduction

The practical usefulness of a method can perhaps only be judged by deploying the method in a real-life application. This chapter assesses the effectiveness of the computational method discussed in the previous chapters as a foreground scene segmentation technique by applying it as a part of a real-life video surveillance approach. The drawbacks of the method when applied to real-life practical scenes are also reported.

Baggage abandoned in public places can pose a serious security threat; such an abandoned bag can be stuffed with explosives that can be detonated by terrorists from a remote location. In this chapter, a two-stage approach is presented to locate abandoned baggage that works on video sequences captured by a single immovable CCTV camera.

At first, foreground objects are segregated from static background objects using brightness and chromaticity distortion parameters estimated in the RGB colour space. This stage of the approach demonstrates how the developed computational model can be modified to segment real targets, excluding the moving shadows, from

the background. The algorithm then locks on to binary blobs that are static and of 'bag' sizes; the size constraints used in the scheme are chosen based on empirical data. Parts of the background frame and current frames covered by a locked mask are then registered if an edge-map based histogram matching (which is the second stage of the approach) yields a positive result, and subsequently tracked using the same method.

It should be mentioned here that the pixels that are part of an edge are marked, to generate an edge-map, using a novel scheme that utilizes four 1-D Laplacian kernels. Tracking is done by constructing histograms based on the intensity images in the sections encompassed by the binary edge maps through the use of a standard histogram matching process. This makes the process broadly illumination invariant. The developed two-stage approach has been tested on the *Imagery Library for Intelligent Detection Systems* (iLIDS) dataset (produced by the Home Office Scientific Development Branch in partnership with Security Service, United Kingdom) and the results obtained are encouraging.

4.2 Chapter organisation

The rest of the chapter is organised as follows: A brief outline of the application in which the two-stage approach has been applied is presented in the following section. Section 4.4 briefly describes the modified computational model that is used in the first stage of the algorithm to segment the true targets (foreground objects and no moving shadows) from the background. Section 4.5 describes the novel edge detector, and talks about edge-map based histogram matching method used in the algorithm. Results are shown and discussed in Section 4.6, and finally conclusions are drawn in Section 4.7.

4.3 A brief outline of the application and the two-stage approach

A lot of work has been reported in the video-surveillance literature on detecting abandoned objects in public places. Some of these works are based on single-camera vision [78], [79], [80] and the rest rely on multi-camera tracking [81], [82], [83], [84], [85]. In general, the authors of these papers have tried to develop fully automatic systems capable of analysing the scene and then pointing out suspicious events like bags being left for long periods of time by their owners. However, a busy scene, change of illumination and strict real-time constraints seriously jeopardize the proper functioning of such systems. It has to be understood that given the current day infrastructure, it is nearly impossible to have a fully automatic intelligent system that can analyse a real-life scene and identify suspicious activities with 100% accuracy.

In this chapter a practical two-stage approach to detect abandoned bag(s) in public areas like underground tube stations is described. The approach serves two purposes: (a) to aid personnel monitoring several screens at the same time to pick up suspicious activities, and (b) to reduce the number of false detections caused by change of illumination. In other words, the algorithm detects and locks onto bag-size stationary blobs in the frames, and subsequently calls for human intervention to take a decision and accordingly some actions.

The first stage of the approach segments foreground objects from the background scene using the computational model discussed in the previous chapters. It has been shown how the computational model can be modified to pick up only the real foreground objects (real-targets), neglecting the shadows (false-targets) from the scene. After the initial scene segmentation, the algorithm locks onto bag-size objects that remain stationary for a few frames. To verify whether the locked blob is an actual target blob or not, an edge-map based histogram matching process is considered as the second stage of the approach. The edges of the image encompassed by the locked binary mask are marked using a novel edge detector that works using four $1-D$ Laplacian kernels. The fact that the histogram of the background frame cov-

ered by the edge map will be different from that of the current frame in the presence of a foreground object is intuitively apparent. The edge detector picks up high frequency spatial components from the images sectioned out by the mask and this makes the process, to a great extent, immune to illumination variations.

4.4 Modified Computational Model

The modified computational model employed works in the RGB colour space and discriminates a pixel as a real-target foreground pixel based on brightness and chromaticity distortions of the pixel with respect to the corresponding pixel in the background frame. The model also exploits the fact that a shadow can be considered as 'a semi-transparent region in the image, which retains a representation of the underlying surface pattern, texture or colour value' to eliminate the shaded regions from the segmented blobs [86], [74], [21], [27]. If $\mathbf{E_{i,j}}^{\dagger}$ is the expected vector ($\mathbf{E_{i,j}} = \left[\bar{r}_{i,j}\ \bar{g}_{i,j}\ \bar{b}_{i,j}\right]$) for the $(i,j)^{th}$ pixel, and if $\mathbf{I_{i,j}}$ is the current vector ($\mathbf{I_{i,j}} = \left[r^c_{i,j}\ g^c_{i,j}\ b^c_{i,j}\right]$) for the same pixel, then brightness distortion, $\Xi_{i,j}$, and chromaticity distortion, $\theta_{i,j}$, are estimated as follows:

$$\Xi_{i,j} = \mid \mathbf{E_{i,j}} \mid - \langle \mathbf{I_{i,j}}, \hat{\mathbf{e}}_{\mathbf{i,j}} \rangle \tag{4.1}$$

$$\theta_{i,j} = \arccos \frac{\langle \mathbf{E_{i,j}}, \mathbf{I_{i,j}} \rangle}{\mid \mathbf{E_{i,j}} \mid\mid \mathbf{I_{i,j}} \mid} \tag{4.2}$$

where, in equation (4.1), $\hat{\mathbf{e}}_{\mathbf{i,j}}$ is the unit vector in the direction of $\mathbf{E_{i,j}}$.
A pixel $\Lambda_{i,j}$ is then treated as a foreground pixel $\Lambda^f_{i,j}$, a shadow pixel $\Lambda^s_{i,j}$ or a background pixel $\Lambda^b_{i,j}$ according to the following rule:

$$\Lambda_{i,j} = \begin{cases} \Lambda^f_{i,j} & if\ (\Xi_{i,j} > \tau_b) \cap (\theta_{i,j} > \tau_\theta) \\ \Lambda^s_{i,j} & if\ (\Xi_{i,j} > \tau_b) \cap (\theta_{i,j} < \tau_\theta) \\ \Lambda^b_{i,j} & otherwise \end{cases} \tag{4.3}$$

where, in equation (4.3), τ_b and τ_θ are the brightness distortion and chromaticity distortion thresholds, respectively; note that the thresholds have been tuned manually.

[†]Note in this chapter pixels are identified on the image plane using both row and column specifiers, unlike points on a lexicographically arranged 1D array as has been done in the previous chapters.

It should also be noted that shadow pixels (false moving targets) are eventually replaced by the corresponding background pixels (a step used in conjunction with equation (4.3)), i.e.:

$$\Lambda_{i,j} = \Lambda^b_{i,j}, \quad if\ \Lambda_{i,j} = \Lambda^s_{i,j} \tag{4.4}$$

The described method is applied to generate binary masks containing all the true foreground targets. The algorithm then locks onto 'bag-sized' stationary blobs on the generated mask. Whether a 'bag-sized' blob is stationary or not is determined by drawing a 11 $pixel \times 11\ pixel$ box around the centroid of the blob and by checking whether or not the centroid remains within the box for 3 consecutive frames. Also note thresholds to identify 'bag-size' blobs have been selected based on empirical data i.e. data collected from the 'Abandoned Baggage' training sequence of the iLIDS dataset. It should also be mentioned that the computational model has been developed to work on 'Lambertian' surfaces; hence, it generates false segmentation results because surfaces in real-life scenes are far from perfect. In addition to this, change of illumination stymies the working of the first stage of the algorithm. Also, studies based on empirical data have shown that brightness distortion estimates play a greater role than chromaticity distortion estimates in segmenting the foreground scene from the background [87]. This makes the first stage of the approach more vulnerable to change of illumination.

To get over the above mentioned problems and so to reduce the number of false detections, a broadly illumination invariant method has been conceived as the second stage of the algorithm. It is known that edges contribute strongly to the high spatial frequency components of an image. In contrast, illumination change in a scene due to omni-directional sources corresponds to the scene-image's low spatial frequency components. Thus if scene-surveillance methods are applied on high frequency component maps, problems due to illumination change can be, by and large, avoided. The second stage employs a method that works using an edge-map based histogram matching process; the method is elaborated in the next section.

4.5 Edge detection and tracking using edge-map dependent histogram matching method

A novel edge detector has been developed that picks up the high frequency components of the image covered by a locked binary-mask. The window of the edge detector comprises 1−D Laplacian kernels in the four directions, as shown in Fig. 4.1. It can be shown that if an edge exits in one of the four directions spanned by the 1−D Laplacian kernels, then the absolute convolution sum in the other directions will return a high value. The maximum value returned by the sub-windows can be checked against a threshold and, if the value is higher than the chosen threshold, the corresponding pixel can be marked as an edge pixel. Going by the same reasoning, it also becomes apparent that in a flat image region the outcome of each of the kernels will be 0, or a value close to 0, and then the output of the edge detector will also be 0. In short, a 5×5 edge detector with four Laplacian sub-windows is scanned through the image covered by the locked binary mask. The maximum of the four absolute convolution sums is determined and, if the maximum value is more than the chosen threshold, the corresponding pixel is marked as an edge pixel.

It should be mentioned here that one of the advantages of using the described edge detector is the fact that the same architecture can be used to detect impulses in an impulse-noise corrupted image. In fact, Zhang and Karim [88] have used a similar structure together with a minimum finding operation to switch the standard square median filter to de-noise images contaminated with impulse noise. It should, however, be pointed out that use of Laplacian kernels results in multi-pixel thick edge lines [2], [1]. In addition to this, it is also known that second differential operators loose the sense of edge direction, and are extremely sensitive to noise [2], [1]. The sensitivity of these operators, though, can prove to be advantageous in some situations; such second differential operators can be used to pick up weak edges in any particular direction. It should also be noted that zero-crossings cannot be utilized to localise the edges, as the filter window is broken up into 4 unidirectional sub-windows and also because of the fact that finding out the maximum of the sub-filter-window output is a non-linear operation. To localise the edge pixels non-maximal suppression algorithms should be applied to the output of the edge detector [89] . Integer

arithmetic has been used in the developed edge detection algorithm to restrict the spread of the edge-lines in some cases.

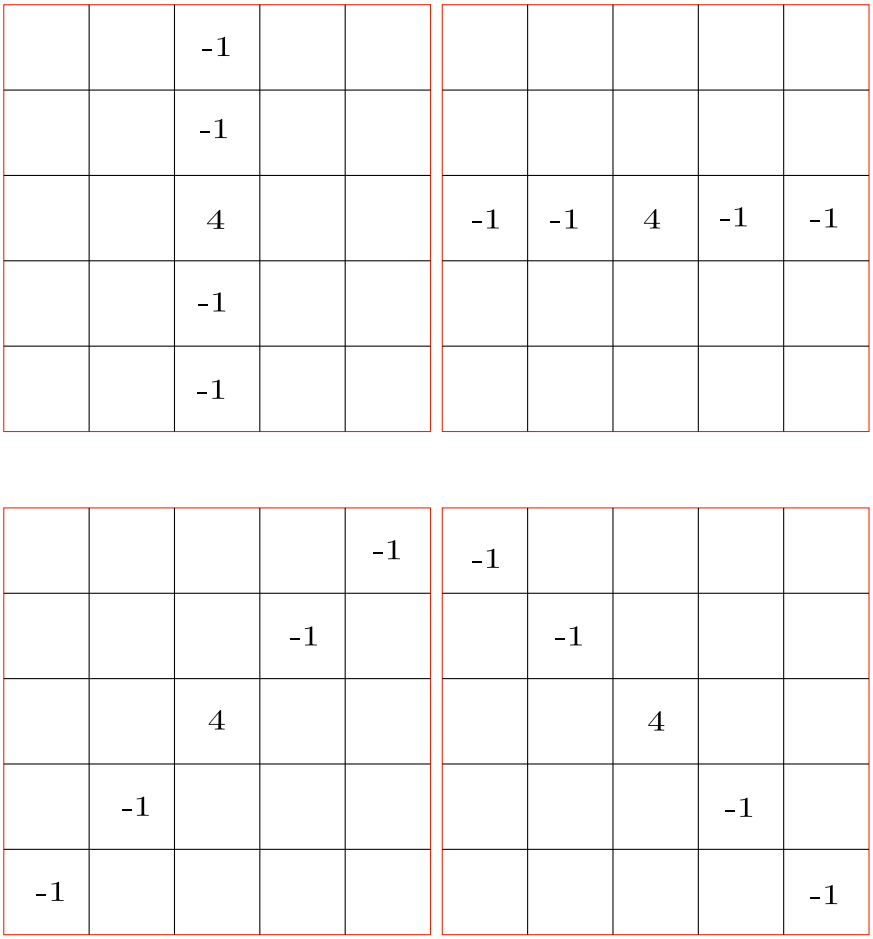


Figure 4.1: The four one directional sub-filter windows of the edge detector.

Edge-maps are generated for those parts of the expected background and current frames that are covered by a locked mask. Edge-map based histograms are constructed and matched for both frames to verify whether the segmented blob actually contains a foreground object or not; this reduces false segmentations that are, by and large, caused by change of illumination. It should be mentioned here that the 'Bhattacharyya matching' method [87] has been used to match the two histograms. In this method, if H_1 and H_2 are the two histograms (with the same number of bins) that are to be matched then the 'Bhattacharyya distance', d_{Bd}, between the

two histograms is calculated as:

$$d_{Bd}(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i)H_2(i)}}{\sqrt{\sum_i H_1(i) \sum_i H_2(i)}}} \tag{4.5}$$

For this matching method, note that low (distance) scores indicate good matches, and high scores bad matches. In the ideal case, a perfect match will return a 0 distance score, and a total mismatch a score of 1. Intensity images, Λ^Is, encompassed by the edge-maps, have been considered while developing the algorithms, and the intensity values for both the frames distributed to 32 equal sized bins.

Note that

$$\Lambda^I_{i,j} \in \{I_p : I_p \in \mathbb{Z} \cap I_p \in [0, 255]\} \tag{4.6}$$

Moreover, note that a distance score of less than 0.35 is taken as a match between the two histograms constructed.

If the histograms are different then the two histograms are registered for future tracking. For every successive frame an edge-map dependent histogram is constructed for the same region covered by the locked mask. If the constructed histogram matches with the histogram of the foreground object, then it is assumed that the object (abandoned bag) is still there. If it matches with that of the background frame, then it is assumed that the object is removed. A fuzzy-state has also been considered in the algorithm. If the edge-map dependent histogram of the current frame neither matches with that of the abandoned bag-sized foreground object nor with that of the background, then it is assumed that the camera-view is obstructed, and human intervention is called for. This step ensures that the approach does not loose the abandoned object even if the view is obstructed with people standing in front of the bag. A first level alarm is generated once the histogram of a foreground object along with the histogram of the background (for the same region) are registered. A more serious level of alarm, that calls for human intervention, is generated if the object is not removed for 60 *sec*.

4.6 Results

As mentioned earlier, the two step approach has been tested on all the sequences of the iLIDS database. Fig. 4.2 shows a typical expected background scene and Fig.

4.3 a commuter sitting on a platform seat, apparently waiting for a train to arrive. The commuter, however, walks out of the scene leaving his bag on the seat adjacent to the one on which he has been sitting. Fig. 4.4 — Fig. 4.6 show three consecutive frames where the bag and the two trains have been segmented using the modified computational model, i.e. the first stage of the approach. The segmented blob for the bag meets the size thresholds and the centroid of the blob remains within its surrounding 11 $pixel \times 11$ $pixel$ box. The edges of the box are then picked up using the edge detector comprising four 1D Laplacian subfilters. A histogram of the intensity image is constructed after that and matched with that of the background. Edge based histograms of the segmented blob of the current frame and of the background have then been registered as the *Bhattacharyya* distance yielded a score of more than 0.35. Fig. 4.7 shows that the bag has been marked as an abandoned object. Note that when an object is marked as abandoned, the corresponding area of the mask is not included while segmenting the foreground scene from the background using the modified computational model in subsequent frames. This is the reason why the small window in Fig. 4.7 shows a binary white region corresponding to the train but not the bag.

Fig. 4.8 shows a moderately busy scene of the same London underground station, and the foreground binary mask of the scene generated using the computational model. Also note that the figure shows the outlook of the software designed and developed for real-time scene surveillance. Fig. 4.9 shows an abandoned bag registered for future tracking after matching the corresponding edge-map based histograms of the current and background frames. A serious level of alarm is generated after tracking the bag for 60 *sec* as shown in Fig. 4.10. Note that this is in accordance with the iLIDS requirements for alarm triggering. Fig. 4.11 shows the camera view obstructed by a commuter standing in front of the bag. As seen in the figure the two stage approach does not lose track of the bag; instead it calls for human intervention and for an appropriate action to be taken.

It should also be mentioned here that the two stage approach has been tested on all the 'Abandoned Baggage' sequences of the iLIDS dataset. Although the results obtained validate the practical usefulness of the approach in most cases and underpins the illumination-change tolerance of the method in some cases, however, it also

generates false alarms. Fig. 4.12 shows one such case where a part of the bag, not abandoned, has been picked up as an abandoned object. Such false alarms bring out another drawback of the computational model that necessitates investigations that involve other methods of foreground scene segmentation (Chapters 5 and 6). Note that even after applying morphological filters, as discussed in Section 2.7 of Chapter 2, discontinuities sometimes remain unfilled between separate blobs demarcated using the brightness and chromaticity distortion parameters. One such disconnected bob has been picked up by the second stage of the approach resulting in the false alarm as shown in Fig. 4.12.

That the modified computational model is efficient in segmenting the foreground scene from the background becomes clear from most of the results obtained, after applying the approach on the iLIDS 'Abandoned Baggage' sequences. However, how good the approach is in neglecting the moving shadows has remained uncertain as these sequences do not contain prominent shadows. To show the efficacy of the computational model in suppressing shadows, two more frames, one from an iLIDS 'Sterile Zone' sequence and another from an iLIDS 'Parked Vehicle' sequence have been included in this chapter. Note from figures Fig. 4.13 and Fig. 4.15 that both frames contain prominent moving shadows. It can be seen from figures Fig. 4.14 and Fig. 4.16 that the modified computational model only segments out the real targets from the background, neglecting the moving shadows, as has been intended while developing the modified model.

Also included in the chapter are two images shown in Fig. 4.17 and Fig. 4.18. The first figure has been generated through the application of the edge detector comprising the four unidirectional Laplacian kernels on the scene showing a London underground tube station platform (Fig. 4.2). The next figure (Fig. 4.18) shows thinning of the multiple pixel thick edge lines of the edge segmented figure, Fig. 4.17; thinning has been done using a standard inbuilt function of MATLAB, '*bwmorph*($\mathbf{X}$ [= Image Vector] , 'thin')' [72]. The two figures have been included to demonstrate how good the performance of the edge detector, described in Section 4.5, is when applied on a typical practical scene.

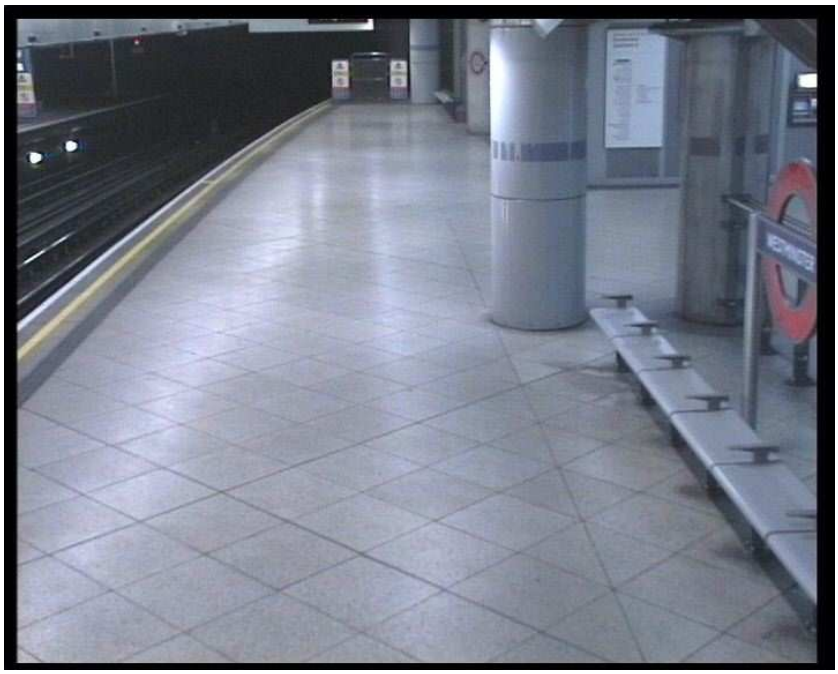


Figure 4.2: Camera view of a London underground tube station platform.

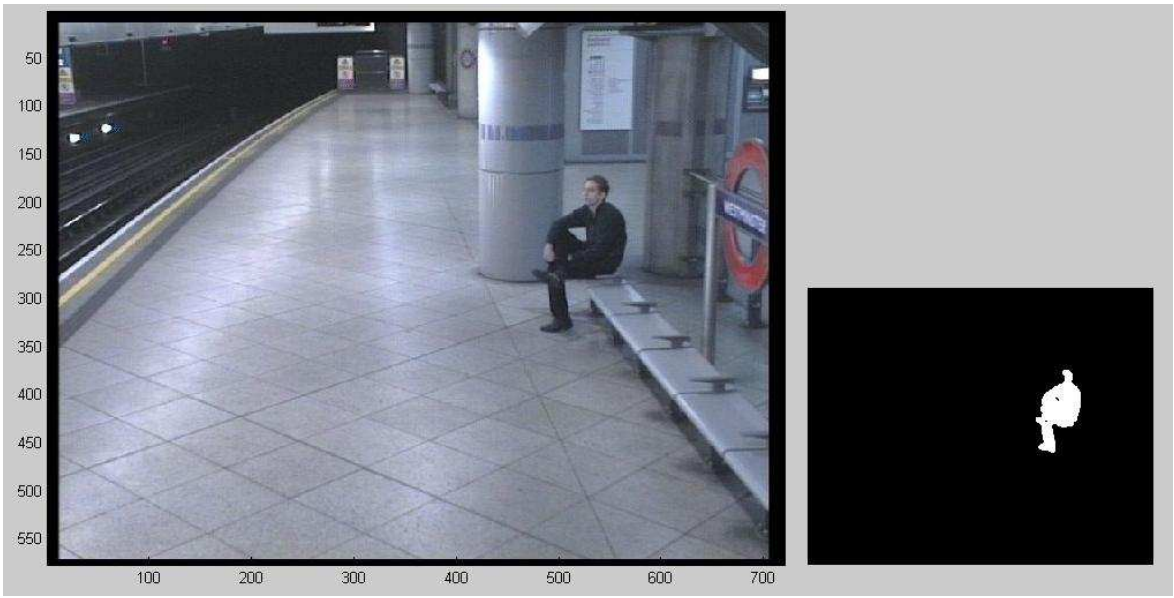


Figure 4.3: A commuter seating on a platform-seat and, apparently, waiting for a train to arrive; the small window shows the binary mask of the foreground scene segmented using the first stage of the algorithm.

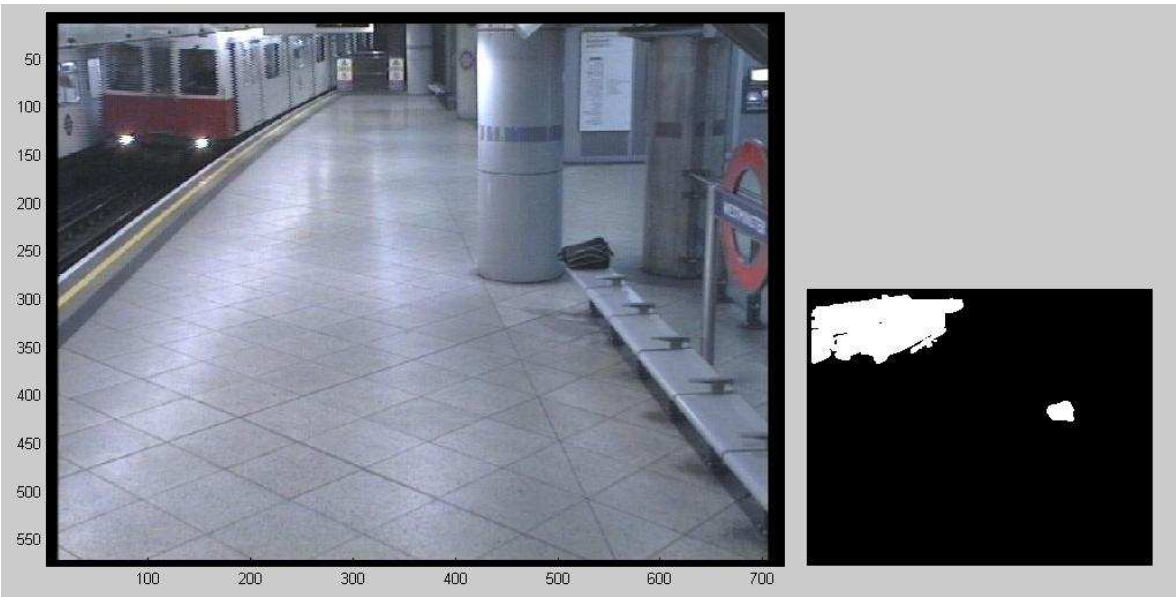


Figure 4.4: The commuter, however, leaves the scene abandoning his bag on a platform-seat. [Size of the blob corresponding to the abandoned bag = [43 $pixel$(height) $\times$ 59 $pixel$(width)] with the centroid of the blob at $(531, 255)$.]

Figure 4.5: The next consecutive frame after the one shown in Fig. 4.4. [Size of the blob corresponding to the abandoned bag = [43 $pixel$(height) $\times$ 59 $pixel$(width)], centroid of the blob = $(531, 255)$.]

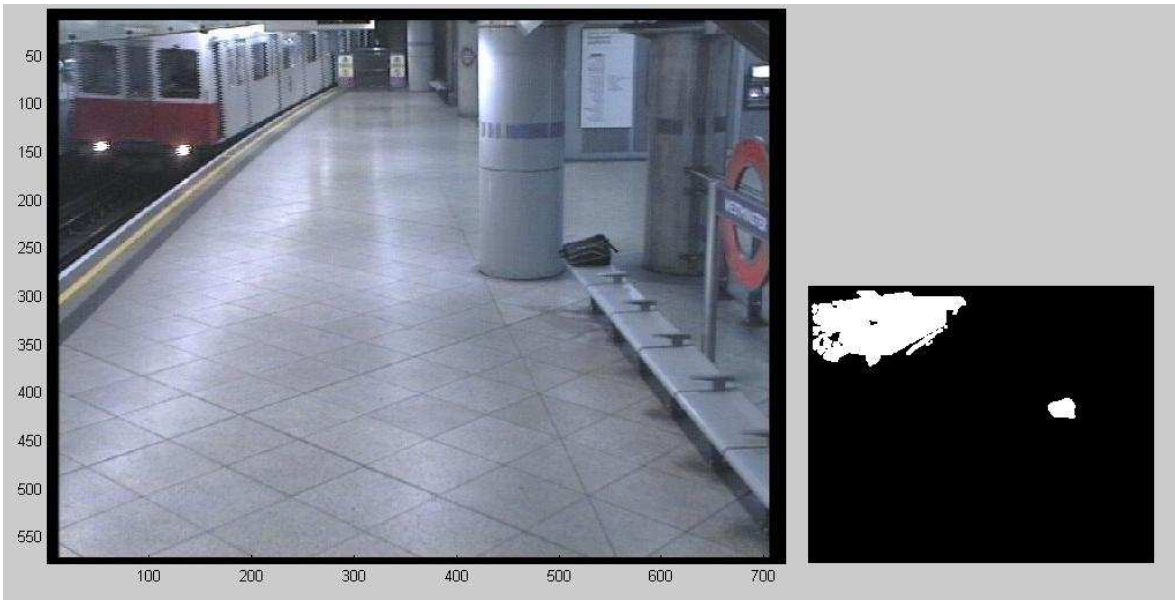


Figure 4.6: The next consecutive frame after the one shown in Fig. 4.5. [Size of the blob corresponding to the abandoned bag = [43 $pixel$(height) $\times$ 59 $pixel$(width)] with the centroid of the blob at $(531, 255)$.]

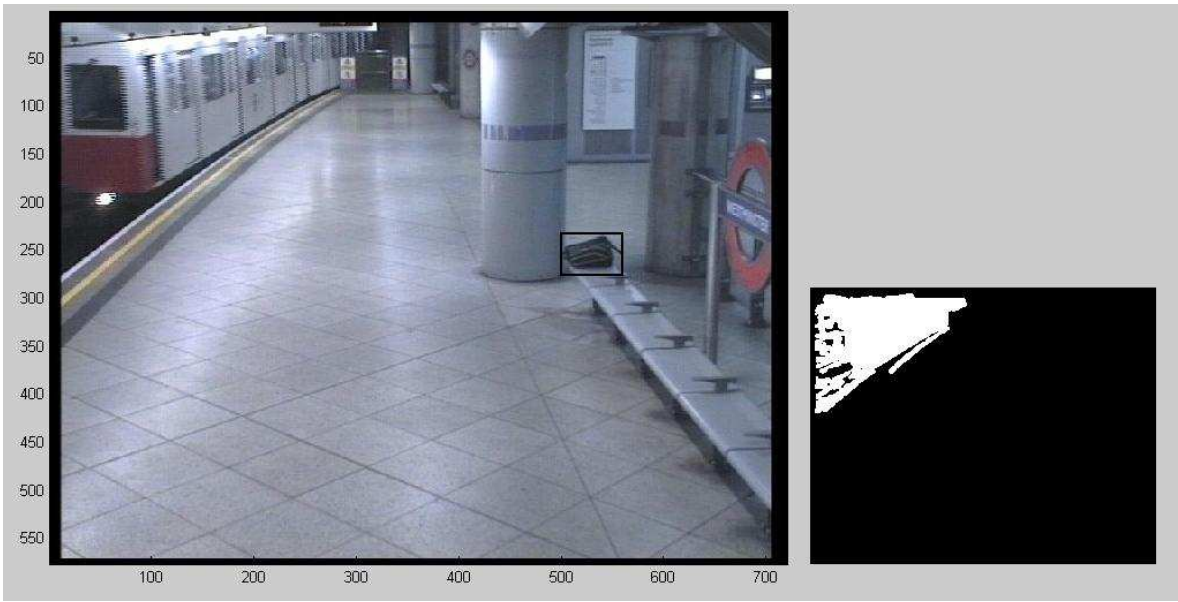


Figure 4.7: The abandoned bag is registered and tracked after applying the edge-map based histogram matching process; note that the area of the frame covering the bag is not included while segmenting subsequent current frames using the modified computational model.

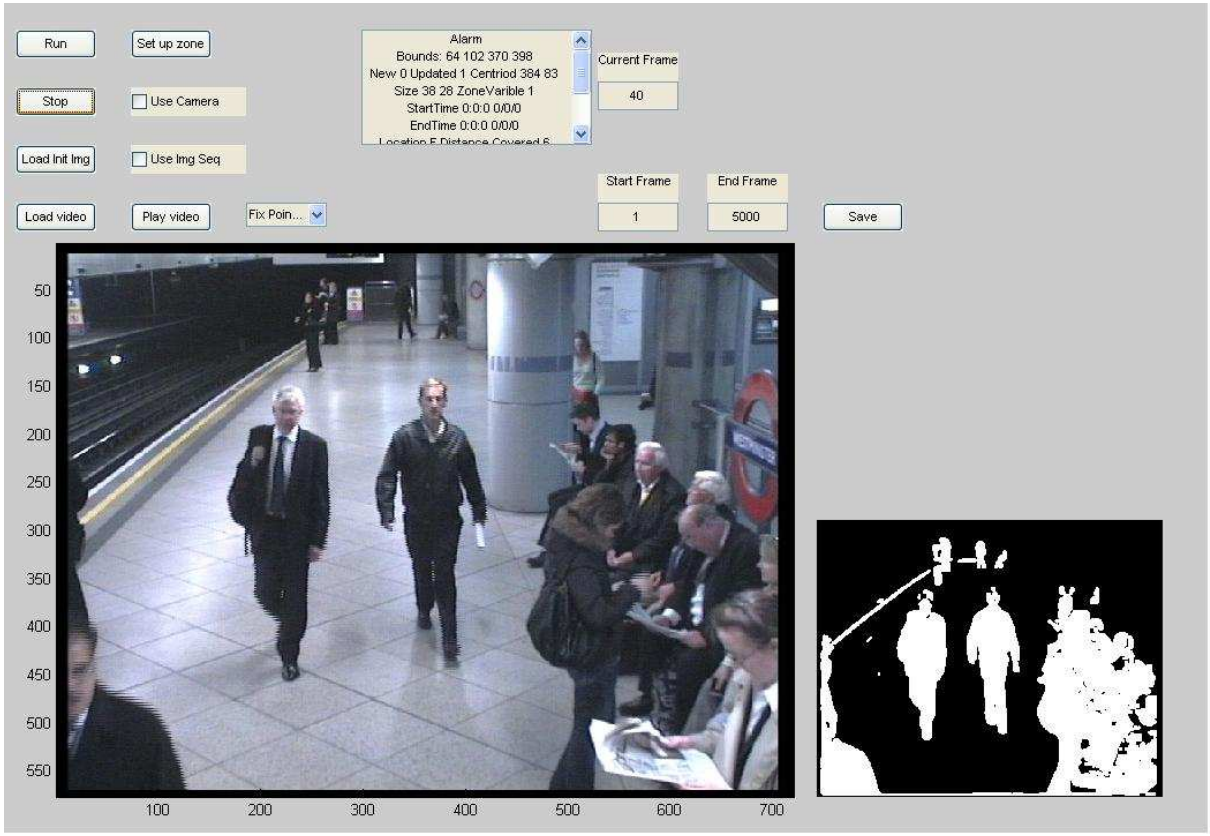


Figure 4.8: A moderately busy scene of a London underground tube station.

Figure 4.9: A static object (abandoned bag) is registered after applying the edge-map based histogram matching process.

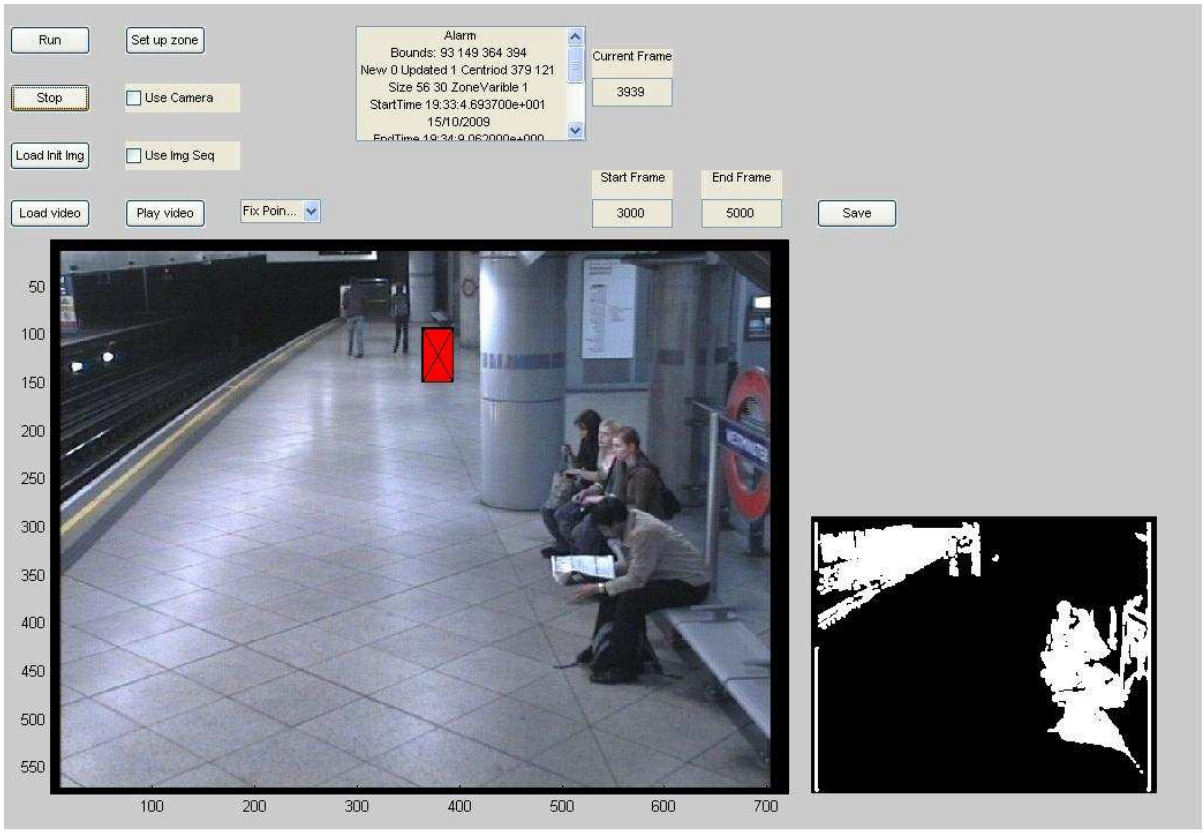


Figure 4.10: A serious alarm is generated after tracking the abandoned bag for 60 seconds.

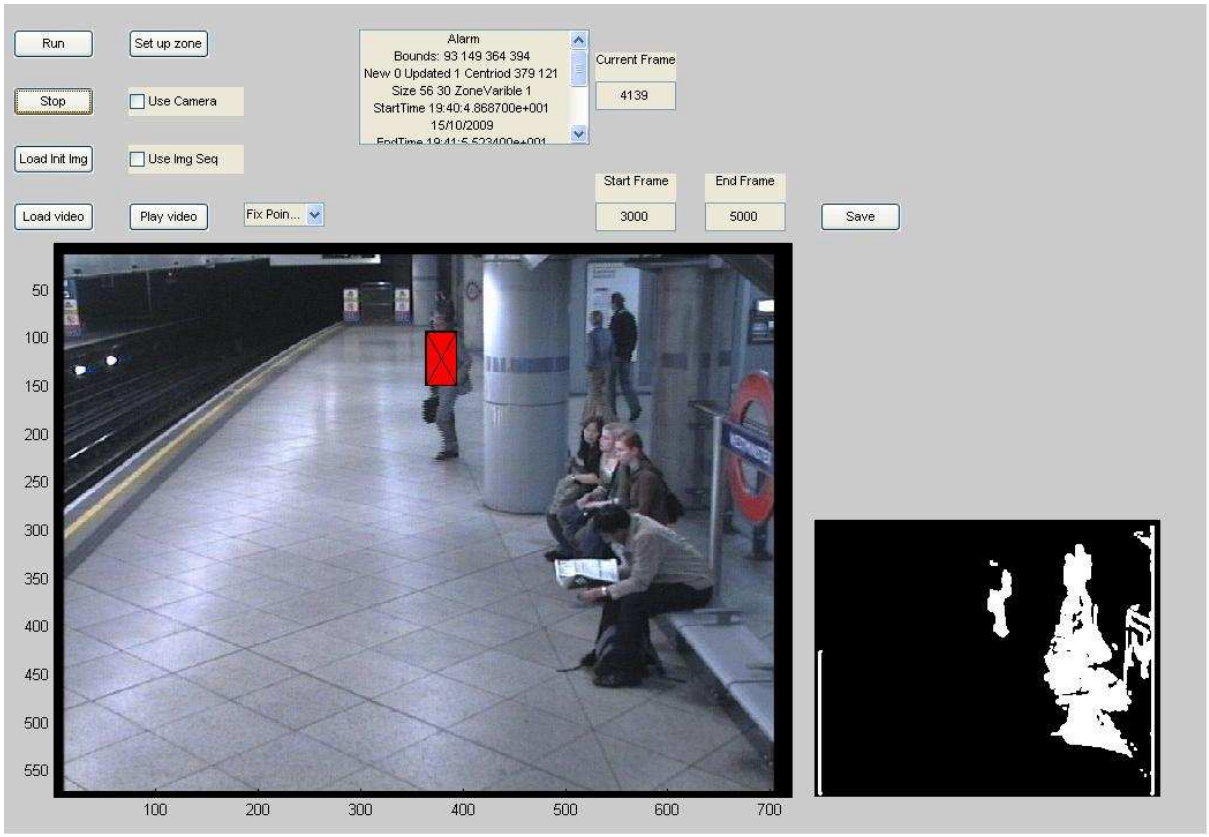


Figure 4.11: The track of the registered abandoned object is not lost even if the view is obstructed; note human intervention is called for whenever the algorithm detects that the camera view is obstructed.

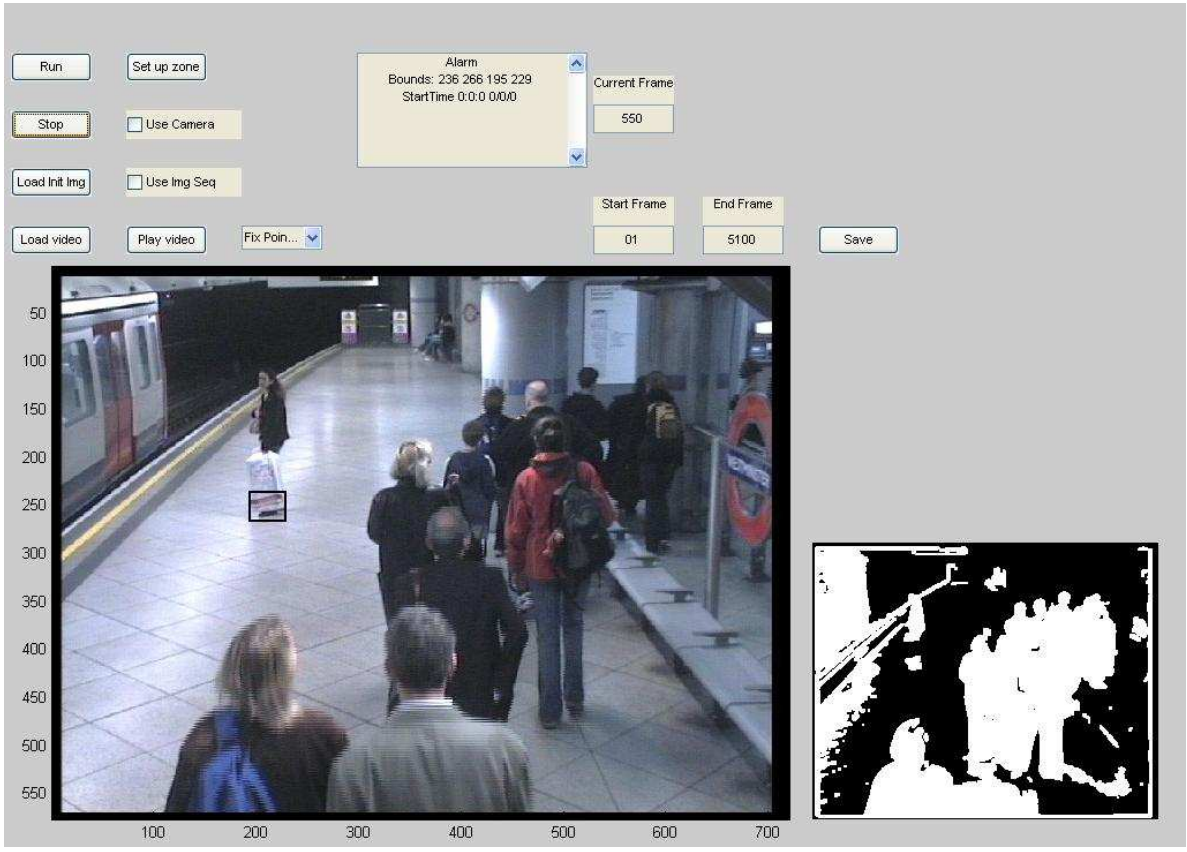


Figure 4.12: A disconnected blob leading to false object registration and tracking.

Figure 4.13: A frame from one of the iLIDS 'Sterile Zone' video sequences.

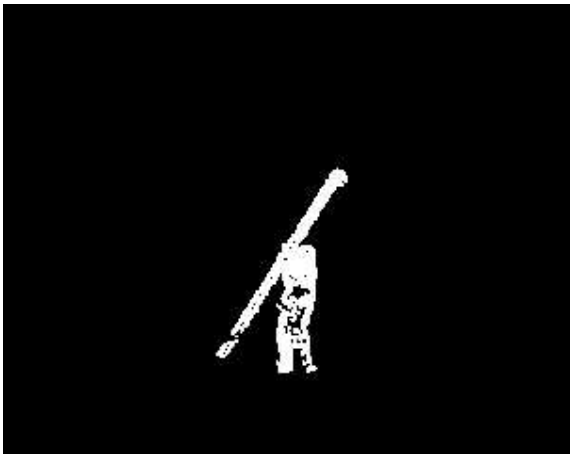


Figure 4.14: The binary mask of the real-target region attained after application of the modified computational method on the frame shown in Fig. 4.13.

4.7 Summary

The chapter describes a practical two-step approach to detect abandoned objects in moderately busy public places. The method first segments a scene using a modified version of the computational model that has been discussed in the previous chapters, and then locks onto stationary bag-size blobs. It then uses an edge-detector to pick up the high frequency components from the background and current frames covered by the locked mask. An edge-map based histogram matching process is then deployed to verify whether the segmented blob actually corresponds to a foreground object (e.g. a left bag) or not. If the answer is affirmative, the registered foreground

Figure 4.15: A frame from one of the iLIDS 'Parked Vehicle' video sequences.

Figure 4.16: The binary mask of the real-target region attained after application of the modified computational method on the frame shown in Fig. 4.15.

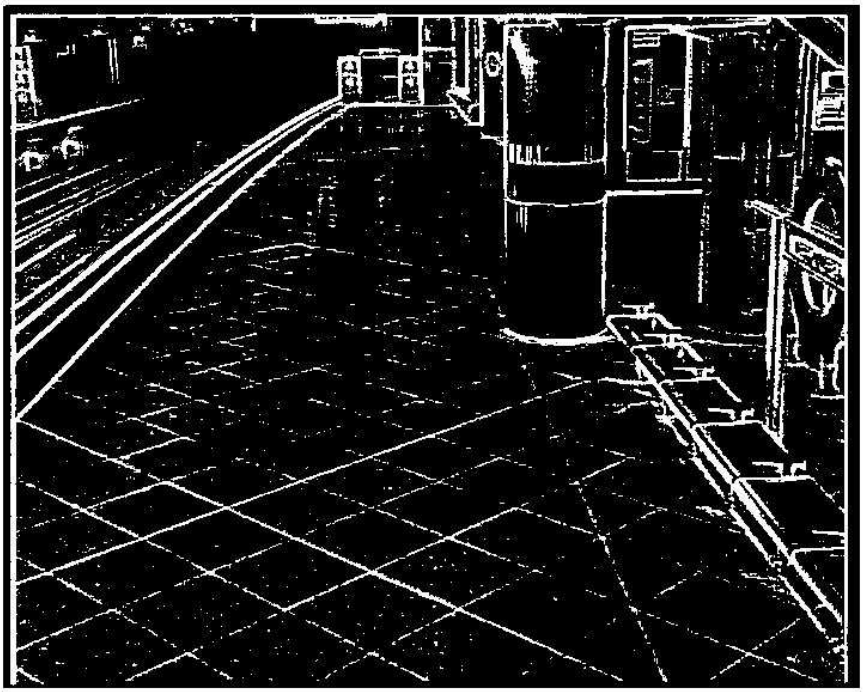


Figure 4.17: The scene shown in Fig. 4.2, edge segmented using the edge detector mentioned in this chapter.

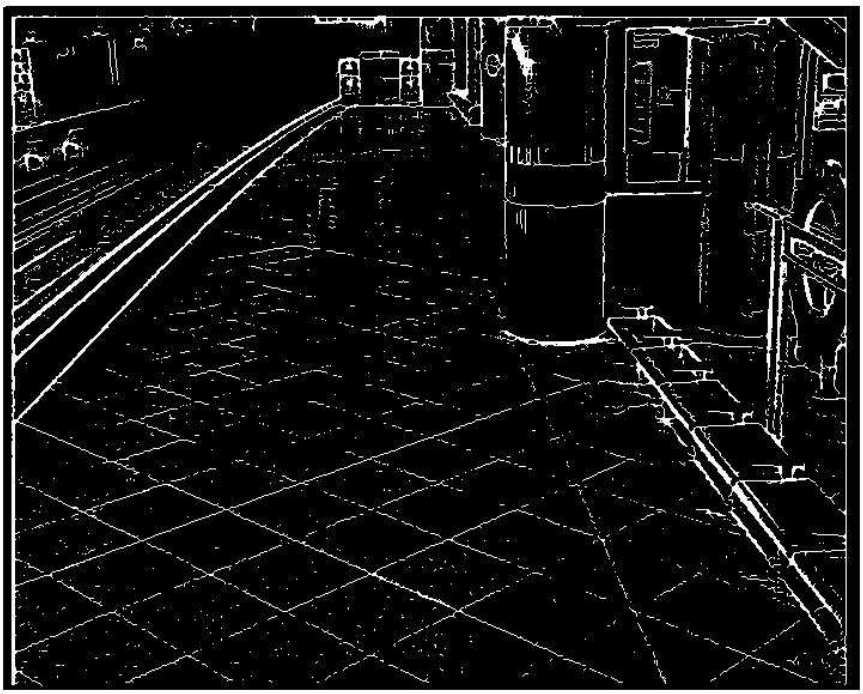


Figure 4.18: The edge lines of the edge segmented scene shown in Fig. 4.17 localised through the use of a non maximal suppression technique (in this case 'thinning').

object is tracked using the edge-map based histogram matching method. Obstructions in camera view are considered as a fuzzy-state and the track of the object is never lost. This stage of the approach makes the overall method broadly illumination invariant and thus reduces the number of false alarms, that other available methods that do not accommodate illumination variance, generate.

It is to be noted that deployment of the modified computational model in a real-life surveillance application has also brought out its drawbacks. It is now clear that the method, if applied on its own, may lead to false segmentation. This is because not only are the surfaces encountered in a real life scene non-Lambertian in nature but also because of the algorithm's inability to cope well with illumination changes in the scene. This means that the method, if employed for the ease of its implementation, has to be supplemented with more robust foreground scene segmentation techniques to prevent any increments in the false detection rate. However, more robust scene segmentation techniques are usually computationally intensive and are thus time-consuming. In the two-stage approach, that has been discussed in this chapter the overall method maintains video-rate processing capability because of the fact that the computationally intensive edge-map based histogram matching process is only applied on the small bag size regions segmented out by the first stage of the approach. It has also been noted that the morphological filters, whose sizes are chosen on an *ad hoc* basis, sometime fail to connect some of the disconnected blobs; this may lead to the generation of false alarms as has been shown for one of the sequences in the previous section.
To overcome the short comings of a standard foreground scene segmentation method like the one developed in the last two chapters of this thesis, more sophisticated and hence robust methods of scene segmentation based on depth information have been investigated in the following chapters.

Chapter 5

MODELING OF AN ACTIVE DEPTH ESTIMATION SYSTEM

5.1 Introduction

In chapters 2 and 3 a computational model has been developed, based on brightness and chromaticity distortion estimates, that can mark and eliminate moving shadows in a scene. A modified version of the model has been outlined in Chapter 4 that explains how to use the model to segment only the real targets, neglecting the moving shadows (false-targets), from a scene. It has been noted in Chapter 4 that though the use of the computational model in a real-life scene surveillance method has its advantages, the model also has its own limitations (Results Section, Chapter 4). To overcome some of these limitations that disrupt the working of the computational model in a real life scene, a more sophisticated method of foreground scene segmentation based on depth estimates is developed in this chapter.

Depth perception, in general, as described in Chapter 1, can be achieved either through a passive method or an active one. This chapter describes the modeling of an active depth estimation arrangement based on the projection of a spot pattern. In short, the purpose of this chapter is two-fold (the second being dependent on the first):

(1) to describe the development of a practical mathematical model of a structured light based depth estimation arrangement.

(2) to achieve, with the help of the mathematical model, an improvement in the

working-range/volume of the system without explicitly encoding the projected spots.

5.2 Chapter organisation

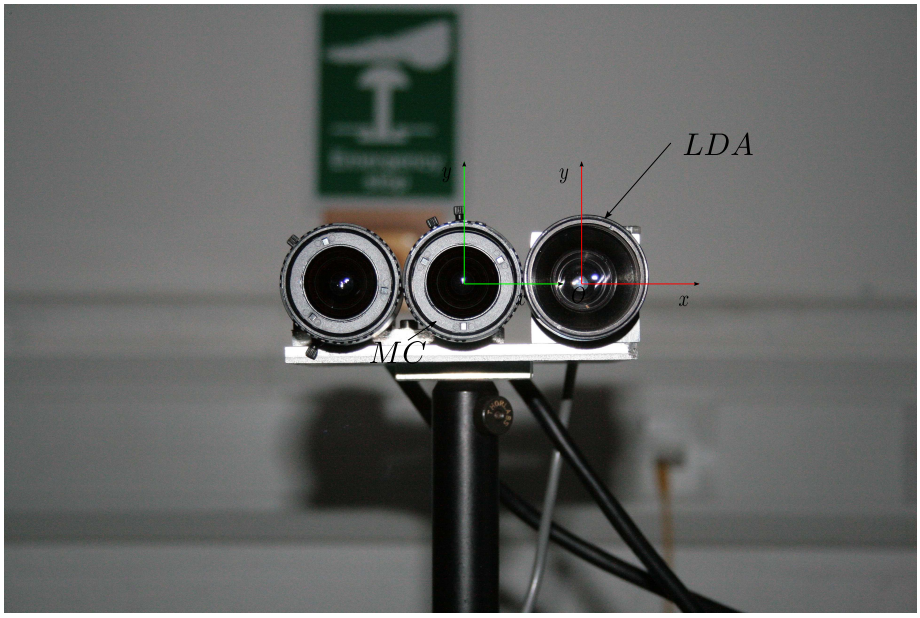
The chapter is organised as follows: Section 5.3 describes the overall set-up used to project the structured pattern and perceive depth. The entire modeling is presented in Section 5.4. How the model parameters can be estimated has been reported in Section 5.5. The same section also demonstrates the functioning of the model and describes a modified method that can also be used to estimate depth. A theory that can be utilised to increase the working volume of the system has been proposed in Section 5.6. Finally, in Section 5.7, the entire chapter has been summarised.

5.3 Experimental arrangement

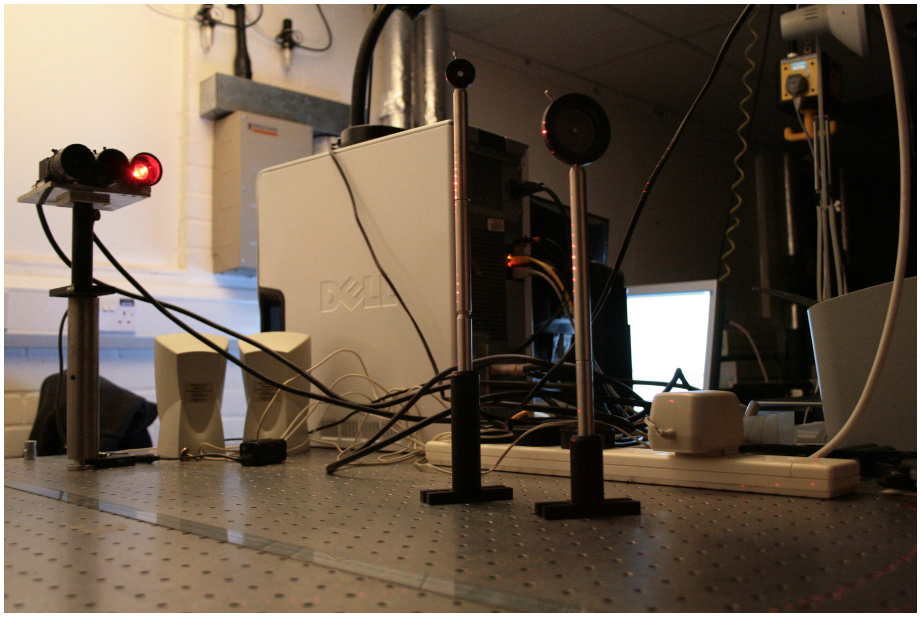
Light from a 100 mW red laser diode (635 nm) is passed through a glass diffractive optical element (DOE) to generate a $2-$D 65×65 array of spots with a constant angular separation. A single fire-wire monochrome camera (POINT GREY RESEARCH, FL2-08S2M), fitted with a $\frac{1}{3}^{"}$ CCD sensor of resolution 1024×768 and placed close to the laser (with geometrical constraints), is used to image the projected pattern. The camera, in turn, is connected to a personal computer (PC); an application program written in C allows the grabbing and saving of the images captured by the camera. The entire arrangement is shown in Fig. 5.1.

5.4 Modeling

A global co-ordinate system (GCS), placed on the centre of the DOE, O, is considered (as shown in Fig. 5.1). The co-ordinate system placed on the centre of the lens, C, of the camera is assumed to have the same orientation as that of the GCS. Thus, the co-ordinates of C with respect to the GCS is can be expressed as $(\varphi_x, \varphi_y, \varphi_z)$. The centre of the image plane/CCD sensor, I, then has the co-ordinates $(\varphi_x, \varphi_y, -f+\varphi_z)$, where f is the focal length of the lens. Note that the entire modeling has been done as a two step process. First, equations to estimate the co-ordinates



(i)



(ii)

Figure 5.1: The active depth sensing system making use of a laser-DOE arrangement (LDA), to project a structured pattern of spots, and a monochrome camera (MC) to image the scene. (i) The centre of the GCS, O, is fixed at the centre of the DOE, and the centre of the other co-ordinate system considered at the principal point of the lens of the camera (note that the $+z$ axis is towards the reader); (ii) a view of the overall set-up (note the colour camera lying to the left of the monochrome camera has not been used).

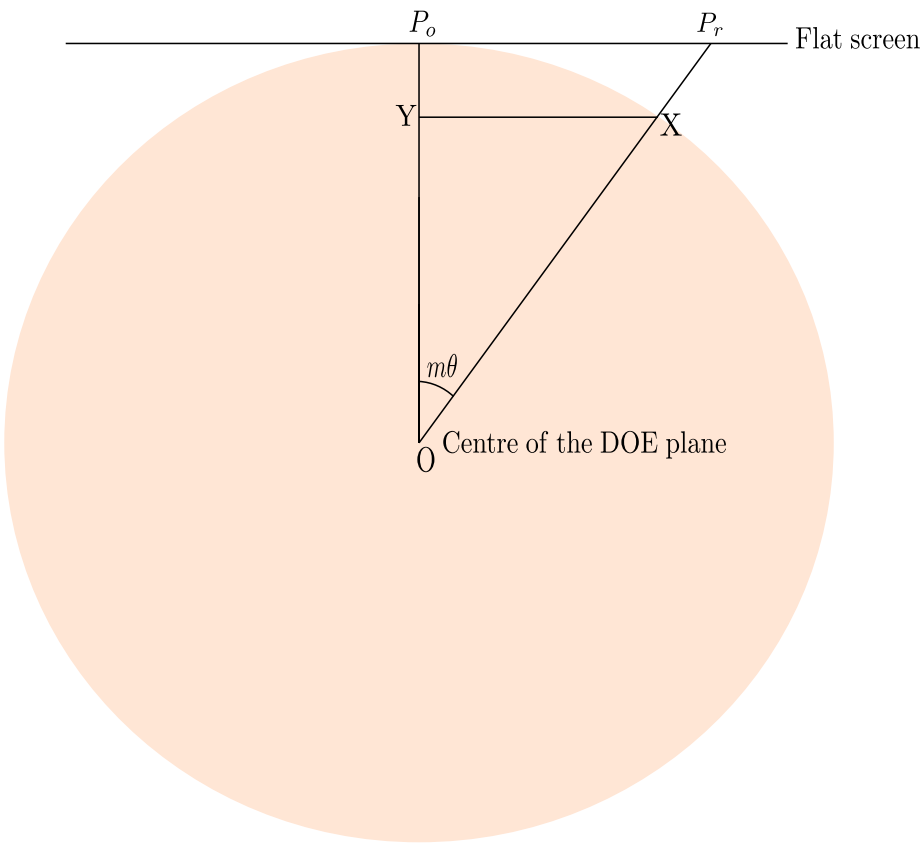


Figure 5.2: Diagram showing the construction made to determine the co-ordinates of the spot, P_r, formed by the incidence of the light ray OXP_r, lying on the central circle-plane, on a flat screen placed at a distance r from the GCS.

of an arbitrary projected spot on a flat screen placed at a distance r from O has been determined. Then, equations to locate the projection of the spot on the image plane of the monochrome camera have been formulated. Thus by defining a mapping between the co-ordinates of an arbitrary laser spot on a screen with that of its projection on the image plane, the model attempts to solve the classical correspondence problem. Also note that the entire modeling has been done based on thin-lens theory.

5.4.1 Determining the co-ordinates of a spot on a flat screen

Determination of the co-ordinates of a spot on a flat screen placed at a distance r, along the z-axis of the GCS, from the laser-DOE arrangement centre, O, has been done by considering a number of slanted concentric circle-planes centred at O and touching the screen along the line: $x = 0, z = r$. Light shafts emanating from the DOE and lying on the slanted circle-planes at regular angular intervals are incident on the flat screen generating a structured pattern. Note that it is imagined that the circle-planes are placed at equal incremental angles, in both directions, with respect to the central circle-plane that lies on the $x - z$ plane of the GCS and touches the flat screen at the point $(0, 0, r)$.

Two angles are now defined that will facilitate development of the mathematical model: the horizontal scan angle, θ, and the vertical scan angle, ϕ. The horizontal scan angle is the angle between two adjacent light shafts lying on the same slanted circle-plane and the vertical scan angle is the angle between the central lines (lines intersecting the flat screen along the line: $x = 0, z = r$) of two adjacent slanted circle-planes.

Let the point where the z-axis of the GCS intersects the screen be P_O. The co-ordinates of P_O are thus $(0, 0, r)$. Co-ordinates of a general point (P_r) on the flat screen formed by the m^{th} light shaft on the central plane in one direction ($-x$ direction shown here) from the central light shaft can be determined by considering the similar triangles $P_O P_r O$ and YXO as shown in Fig. 5.2. P_r is thus given as $(-r(\tan(m\theta)), 0, r)$. [Note that $m\theta$ denotes the angle between the m^{th} light shaft and the central light shaft both lying on the same circle-plane; in our case m can be

$0, \pm1, \pm2, \ldots, \pm32]$. A more general case of that of determining the co-ordinates of

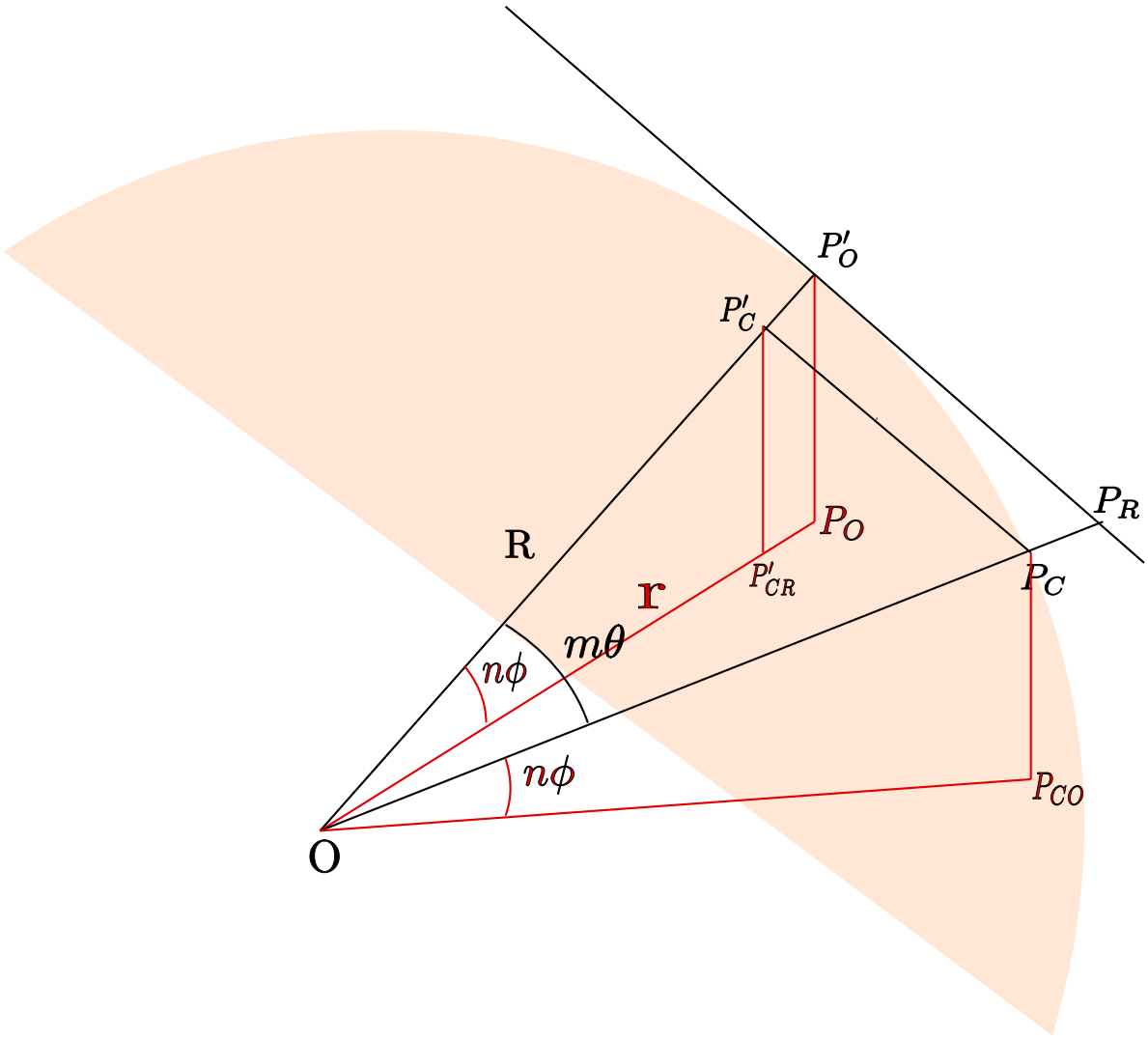


Figure 5.3: Diagram depicting the geometry used to determine the co-ordinates of any arbitrary spot, P_R, of the structured pattern projected on a flat screen placed at a distance of r from the GCS.

any arbitrary point on the screen (not lying on the central circle-plane) is done by considering the slanted circle as shown in Fig. 5.3.

Consider the radius of the slanted circle to be R and the angle between the two central lines OP'_O and OP_O (lying on the central plane) be $n\phi$ [n, in our case can take the values: $0, \pm1, \pm2, \ldots, \pm32$]. The co-ordinates of P'_O, determined considering the triangle $P'_O O P_O$ are $(0, R\sin(n\phi), r)$. Noting from the same triangle that $R = \frac{r}{\cos(n\phi)}$, we have the co-ordinates of P'_O as $(0, r\tan(n\phi), r)$.

Let $P_R(x_{PR}, y_{PR}, z_{PR})$ be an arbitrary laser spot on the flat-screen; this point has been formed by the light shaft originating at O and passing through $P_C(x_C, y_C, z_C)$ intersecting the screen at P_R. The equation of the line describing the light shaft can be written as :

$$P_R = P_C + \alpha(O - P_C) \tag{5.1}$$

This can be expressed in matrix form as:

$$\begin{bmatrix} x_{PR} \\ y_{PR} \\ z_{PR} \end{bmatrix} = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} + \alpha \begin{bmatrix} -x_C \\ -y_C \\ -z_C \end{bmatrix} \tag{5.2}$$

A triangle $P_C P_C^{'} O$ is constructed on the slanted circle-plane by dropping a perpendicular from P_C to the line $OP_O^{'}$; the perpendicular intersects the line $OP_O^{'}$ at $P_C^{'}$. From the triangle, we get the abscissa of the point P_C as $-R\sin(m\theta)$ [note $-x$ direction chosen]. The y-ordinate of the point P_C is found by dropping a perpendicular from P_C to the central circle-plane; the perpendicular intersects the central circle-plane at P_{CO}. From the triangle $P_C O P_{CO}$ we get the y-ordinate of P_C as $R\sin(n\phi)$. Finally, the z-ordinate of P_C is obtained from the two right-angled triangles $P_C P_C^{'} O$ and $P_C^{'} O P_{CR}^{'}$ as $R\cos(m\theta)\cos(n\phi)$. [Note that from the triangle $P_C P_C^{'} O$ we get $OP_C^{'}$ as $R\cos(m\theta)$ and then from the triangle $P_C^{'} O P_{CR}^{'}$ we get $OP_{CR}^{'}$ (the z-ordinate of P_C) as $R\cos(m\theta)\cos(n\phi)$.]

After plugging in the values in equation (5.2) we get:

$$\begin{bmatrix} x_{PR} \\ y_{PR} \\ z_{PR} \end{bmatrix} = \begin{bmatrix} -R\sin(m\theta) \\ R\sin(n\phi) \\ R\cos(m\theta)\cos(n\phi) \end{bmatrix} + \alpha \begin{bmatrix} R\sin(m\theta) \\ -R\sin(n\phi) \\ -R\cos(m\theta)\cos(n\phi) \end{bmatrix} \tag{5.3}$$

Making use of the fact that $z_{PR} = r$, we get:

$$\begin{aligned} r =& R\cos(m\theta)\cos(n\phi) + (-\alpha)R\cos(m\theta)cos(n\phi) \\ =& r\cos(m\theta) - \alpha r\cos(m\theta) \quad [\because R = \frac{r}{\cos(n\phi)}] \end{aligned} \tag{5.4}$$

$$\therefore \alpha = \frac{\cos(m\theta) - 1}{\cos(m\theta)} \tag{5.5}$$

Now, from equation (5.3) we get:

$$x_{PR} = -R\sin(m\phi) + \alpha R\sin(m\phi) \tag{5.6}$$

Using equation (5.5) and the fact $R = \frac{r}{\cos(n\phi)}$ we get:

$$\begin{aligned} x_{PR} =& \frac{-r\sin(m\theta)}{\cos(n\phi)} + \left(\frac{\cos(m\theta)-1}{\cos(m\theta)}\right)\frac{r\sin(m\theta)}{\cos(n\phi)} \\ =& \left(-\frac{r\tan(m\theta)}{\cos(n\phi)}\right) \\ =& \frac{-r\tan(\theta_T)}{\cos(\phi_T)} \text{ where } \theta_T = m\theta, \text{ and } \phi_T = n\phi \end{aligned} \tag{5.7}$$

Similarly,

$$y_{PR} = R\sin(n\phi) - \alpha R\sin(n\phi) \tag{5.8}$$

Therefore,

$$\begin{aligned} y_{PR} =& r\tan(n\phi) - \left(\frac{\cos(m\theta)-1}{cos(m\theta)}\right) r\tan(n\phi) \\ =& \frac{r\tan(n\phi)}{\cos(m\theta)} \\ =& \frac{r\tan(\phi_T)}{\cos(\theta_T)} \end{aligned} \tag{5.9}$$

The above equations [(5.7), (5.9)] bring to light the symmetric nature of the generated pattern.

Note that the signs of x_{PR} and y_{PR}, for an arbitrary spot in the first quadrant, are in accordance with the global co-ordinate system defined. Also note that from this point onwards identification of a spot will be made by specifying its position on the screen (either co-ordinates of the spot will be mentioned explicitly or through the following information: $\langle Q, d, \theta_T(or\ m), \phi_T(or\ n)\rangle$ where Q indicates the quadrant where the spot belongs, i.e., $Q \in \{1(First), 2(Second), 3(Third), 4(Fourth)\}$ and d the distance of the spot along the $z-$ axis from O) (refer to Fig. 5.4); also we consider spots lying on the first quadrant $[\langle 1, d, \theta_T, \phi_T\rangle]$ and the central spot only, unless otherwise stated.

The validity of the proposed equations (co-ordinates of any arbitrary spot on the flat-screen) has been verified by simulating the pattern using the derived equations in MATLAB. The horizontal scan angle (θ) and the vertical scan angle (ϕ) have been taken as 0.7° and r as 1.

Fig. 5.5 shows the simulated pattern in MATLAB, and Fig. 5.6 a camera-shot of the actual pattern. The simulated pattern resembles the actual pattern if we ignore the higher diffraction orders of the DOE.

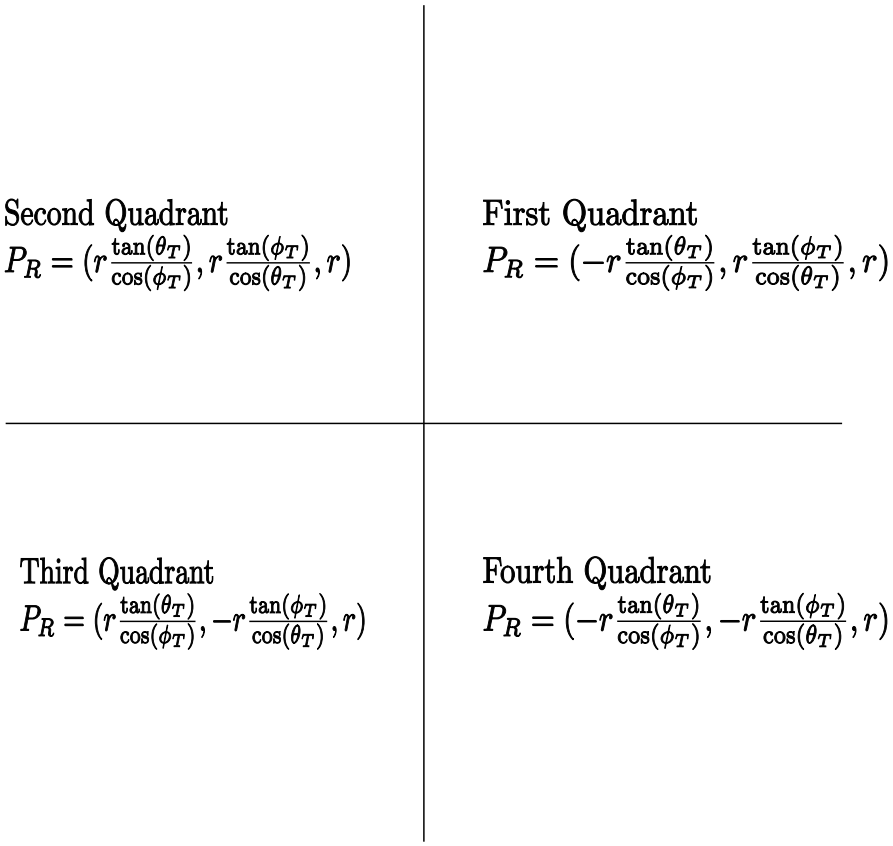


Figure 5.4: General quadrant-specific co-ordinates of arbitrary spots of the pattern projected on a flat screen placed at a distance r from the centre of the GCS.

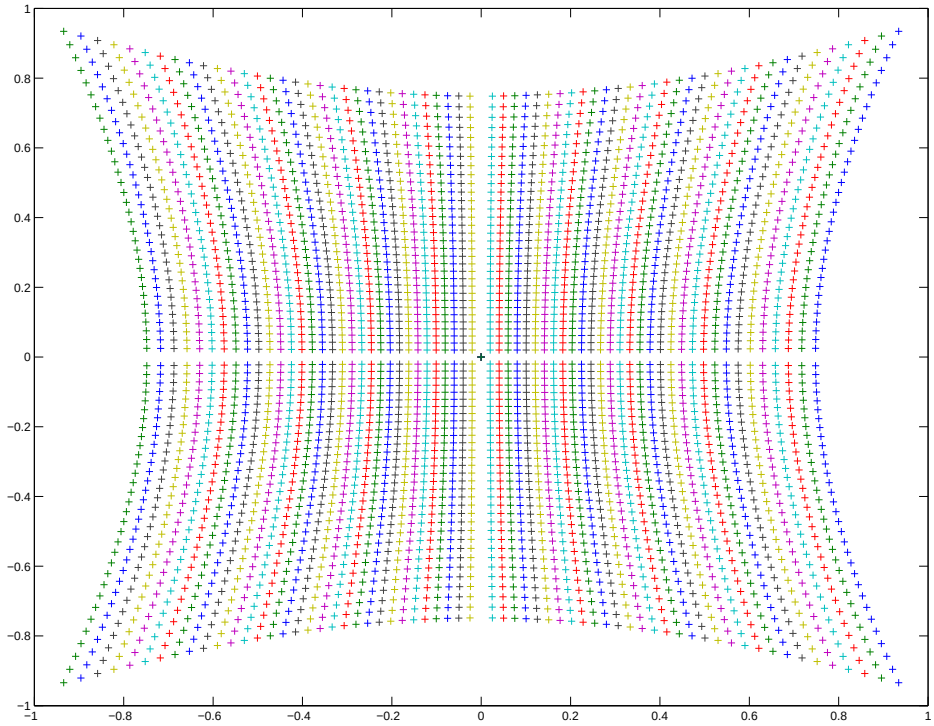


Figure 5.5: The structured pattern simulated in MATLAB using the model equations.

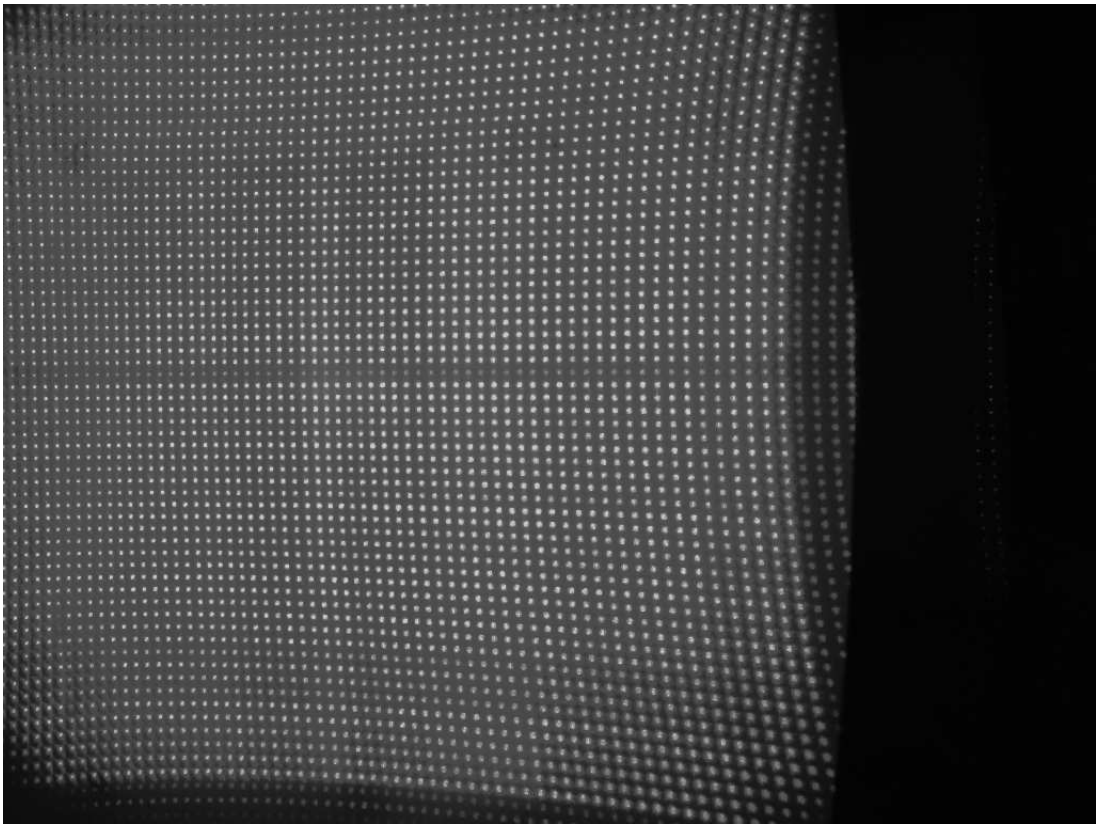


Figure 5.6: Camera-shot of the projected structured pattern.

5.4.2 Determination of the co-ordinates of a spot on the image plane

As has been mentioned earlier, the camera is placed close to the laser arrangement with geometrical constraints and the entire modeling is done based on thin-lens theory. The co-ordinates of the principal point of the lens, C, with respect to the GCS are taken as $(\varphi_x, \varphi_y, \varphi_z)$. The principal point of the image plane, I, is thus $(\varphi_x, \varphi_y, -f + \varphi_z)$.

It has been ensured, whilst building the system, that the principal point of the lens has relatively smaller y- and z- shifts compared to its x-shift.

Now a light ray from any arbitrary spot on the screen, say P_R (x_{PR}, y_{PR}, z_{PR}) will pass through C and will intersect the image plane at the point P_I (x_{PI}, y_{PI}, z_{PI}).

So, P_I can be expressed as:

$$P_I = C + \beta \left[P_R - C \right] \tag{5.10}$$

This can be expressed in matrix form as:

$$\begin{bmatrix} x_{PI} \\ y_{PI} \\ z_{PI} \end{bmatrix} = \begin{bmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{bmatrix} + \beta \begin{bmatrix} x_{PR} - \varphi_x \\ y_{PR} - \varphi_y \\ z_{PR} - \varphi_z \end{bmatrix} \tag{5.11}$$

Putting the values of x_{PR} and y_{PR} from the equations (5.7) and (5.9) and using the fact that $z_{PR} = r$ we get:

$$\begin{bmatrix} x_{PI} \\ y_{PI} \\ z_{PI} \end{bmatrix} = \begin{bmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{bmatrix} + \beta \begin{bmatrix} \frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x \\ \frac{r\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y \\ r - \varphi_z \end{bmatrix} \tag{5.12}$$

So, we have:

$$\beta = \begin{bmatrix} \frac{x_{PI} - \varphi_x}{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x} \end{bmatrix} = \begin{bmatrix} \frac{y_{PI} - \varphi_y}{\frac{r\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y} \end{bmatrix} = \begin{bmatrix} \frac{z_{PI} - \varphi_z}{r - \varphi_z} \end{bmatrix} \tag{5.13}$$

Considering the image plane is one focal length distance behind C we have:

$$z_{PI} = -f + \varphi_z \tag{5.14}$$

From equations (5.13) and (5.14) we get x_{PI} and y_{PI} as:

$$x_{PI} = \begin{bmatrix} \frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z} \end{bmatrix} (-f) + \varphi_x \tag{5.15}$$

$$y_{PI} = \begin{bmatrix} \frac{\frac{r\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y}{r - \varphi_z} \end{bmatrix} (-f) + \varphi_y \tag{5.16}$$

It should be noted that equations (5.15) and (5.16) help us to plot the movement of a spot on the image plane with change of r.

The depth of an object point can be estimated by rearranging either (5.15) or (5.16)as:

$$r = \begin{bmatrix} \frac{f\varphi_x + (x_{PI} - \varphi_x)\varphi_z}{(x_{PI} - \varphi_x) - \frac{f\tan(\theta_T)}{\cos(\phi_T)}} \end{bmatrix} \tag{5.17}$$

or :

$$r = \begin{bmatrix} \frac{f\varphi_y + (y_{PI} - \varphi_y)\varphi_z}{(y_{PI} - \varphi_y) + \frac{f\tan(\phi_T)}{\cos(\theta_T)}} \end{bmatrix} \tag{5.18}$$

5.4.3 Special cases

Giving the principal point of the camera lens, C, only a lateral shift with respect to the GCS results in some interesting effects which are discussed in this subsection in a sequential manner.

If there are no y- and z- shifts of the camera with respect to O of the GCS (i.e. $\varphi_y = 0$, $\varphi_z = 0$), then equation (5.16) can be further simplified as:

$$y_{PI} = \left(\frac{\tan(\phi_T)}{\cos(\theta_T)}\right)(-f) \tag{5.19}$$

In this particular case, since y_{PI} is independent of r and is constant for a particular projected spot, it becomes apparent that there would not be any y-shift of the spot on the image plane with change of r. This, in turn, means that the spots will only move in the x-direction on the image plane ($-ve$ x direction if the object is moved away from O, and $+ve$ x direction if the object comes closer to O) when this special case is considered.

Moreover, if φ_z is neglected i.e. $(\varphi_z \approx 0)$, equation (5.17) can be simplified to:

$$r = \frac{f\varphi_x}{(x_{PI} - \varphi_x) + \frac{f\tan(\theta_T)}{\cos(\phi_T)}} \tag{5.20}$$

Also equation (5.15) can be written as:

$$\begin{aligned} x_{PI} &= \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r}\right](-f) + \varphi_x \\ &= \left[\frac{\tan(\theta_T)}{\cos(\phi_T)} + \frac{\varphi_x}{r}\right](f) + \varphi_x \end{aligned} \tag{5.21}$$

Now, if the screen or an object is moved to a distance r_1 (the corresponding point on the image plane is x_{PI1}) from O, then the x-shift of a spot on the image plane can be expressed as:

$$\begin{aligned} x_{PI1} - x_{PI} &= \left[\frac{\tan(\theta_T)}{\cos(\phi_T)} + \frac{\varphi_x}{r_1}\right](f) - \left[\frac{\tan(\theta_T)}{\cos(\phi_T)} + \frac{\varphi_x}{r}\right](f) \\ &= (\varphi_x f)\left[\frac{1}{r_1} - \frac{1}{r}\right] \\ &= (\varphi_x f)\left[\frac{r - r_1}{r r_1}\right] \end{aligned} \tag{5.22}$$

The above equation (5.22) suggests that the x-shift is more when the object is close to O, than when it is far off. A similar equation can also be derived for the y-shift of

a spot on the image plane, for $\varphi_z = 0$, with change of the screen or object distance from O of the GCS.

5.4.4 Finding out the gradient of the path a spot would follow on the image plane

The aim of this section is to determine the path the projection of an arbitrary laser spot will follow on the image plane. Consequently, it may help in spot tracking on the camera image plane and make spot movement manoeuvring possible.

It has been mentioned earlier that the two equations [(5.15) and (5.16)] give us the x co-ordinate and the y co-ordinate of a spot with change of r. From now on, we will treat equation (5.15) as the 'x-generator' function and equation (5.16) as the 'y-generator' function of a spot on the image plane. To determine the gradient of the path a spot will follow on the image plane with change of r both equations (5.15) and (5.16) are differentiated with respect to r:

$$\begin{aligned}\frac{dy_{PI}}{dr} =& \frac{d}{dr}\left[\left(\frac{r\frac{\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y}{r-\varphi_z}\right)(-f) + \varphi_y\right] \\ =& \left[\frac{r\dfrac{\tan(\phi_T)}{\cos(\theta_T)} - \varphi_z\dfrac{\tan(\phi_T)}{\cos(\theta_T)} - r\dfrac{\tan(\phi_T)}{\cos(\theta_T)} + \varphi_y}{(r-\varphi_z)^2}\right](-f)\end{aligned}$$

Therefore,

$$dy_{PI} = \left[\frac{\varphi_z\frac{\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y}{(r-\varphi_z)^2}\right](f)\,dr \tag{5.23}$$

Similarly, we get:

$$\begin{aligned}\frac{dx_{PI}}{dr} =& \frac{d}{dr}\left[\left(\frac{-r\frac{\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r-\varphi_z}\right)(-f) + \varphi_x\right] \\ =& \left[\frac{-r\dfrac{\tan(\theta_T)}{\cos(\phi_T)} + \varphi_z\dfrac{\tan(\theta_T)}{\cos(\phi_T)} + r\dfrac{\tan(\theta_T)}{\cos(\phi_T)} + \varphi_x}{(r-\varphi_z)^2}\right](-f)\end{aligned}$$

$$\therefore dx_{PI} = -\left[\frac{\varphi_z\frac{\tan(\theta_T)}{\cos(\phi_T)} + \varphi_x}{(r-\varphi_z)^2}\right](f)\,dr \tag{5.24}$$

Combining equations (5.23) and (5.24) we get:

$$\begin{aligned}\frac{dy_{PI}}{dx_{PI}} &= -\left[\frac{\varphi_z \frac{\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y}{\varphi_z \frac{\tan(\theta_T)}{\cos(\phi_T)} + \varphi_x}\right] \\ &= -\left[\frac{\varphi_z \tan(\phi_T)\cos(\phi_T) - \varphi_y \cos(\theta_T)\cos(\phi_T)}{\varphi_z \tan(\phi_T)\cos(\phi_T) + \varphi_x \cos(\theta_T)\cos(\phi_T)}\right] \\ &= -\left[\frac{\varphi_z \sin(\phi_T) - \varphi_y \cos(\theta_T)\cos(\phi_T)}{\varphi_z \sin(\theta_T) + \varphi_x \cos(\theta_T)\cos(\phi_T)}\right] \end{aligned} \tag{5.25}$$

It is evident from equation (5.25) that the gradient (in which direction a spot will move on the image plane) depends on the translational matrix, $\mathbf{T}$ $(= [\varphi_x\ \varphi_y\ \varphi_Z]^T)$, that specifies the shifts of the principal point of the camera lens, C, in the three orthogonal directions with respect to the GCS; it also depends upon some of the spot location specifiers: the quadrant where it belongs, θ_T, and on ϕ_T i.e. $\langle Q,\ :,\ \theta_T,\ \phi_T \rangle$. It is apparent from equation (5.25) that the gradient assumes a constant value when there is either no or a very small z-displacement of the principal point of the lens with respect to the GCS. Neglecting φ_z we get from equation (5.25):

$$\frac{dy_{PI}}{dx_{PI}} = \frac{\varphi_y}{\varphi_x} \tag{5.26}$$

Equation (5.26) makes apparent the fact that all the points of a quadrant will move in the same direction when $\varphi_z = 0$; the direction solely depends upon the y-shift and the x-shift of C with respect to O and not on θ_T and ϕ_T. In addition to these points, it also becomes clear from equation (5.26) that the spot movement can be guided by giving the principal point of the camera lens specific shifts with respect to the GCS. For example, if the principal point is given equal shifts in the x and y directions i.e. $\varphi_x = \varphi_y$, then $\frac{dy_{PI}}{dx_{PI}} = 1$; this means that all the spots will move at an angle of 45° on the image plane with respect to the GCS. This spot movement guiding factor can be efficiently used to increase the working range/volume of the system as will be discussed later.

If the principal point of the camera lens is only given a lateral shift with respect to O i.e. $\varphi_y = 0$, then $\frac{dy_{PI}}{dx_{PI}} = 0$; this means all the spots independent of their positions on the flat screen or on the image plane will move horizontally with the shift of the target. This is in full agreement with equation (5.16) that shows that

the 'y-generator' function is independent of r if $\varphi_z = \varphi_y = 0$. Similarly, if $\varphi_x = 0$, then $\dfrac{dy_{PI}}{dx_{PI}} \longrightarrow \infty$; this means all the spots on the image plane, independent of their positions, will move vertically with the shift of the target.

5.4.5 Magnification

The aim of this section is to find out whether the scale change of an object dimension is linear or not when the object is moved forward or backward with respect to O of the GCS. This is done to determine whether additional non-linearity has to be taken into account when using the LDA to project the pattern instead of a standard digital projector.

Let the co-ordinates of two arbitrary points on the screen placed at a distance r from O be P_{R1} and P_{R2} (Fig. 5.7).

$$P_{R1} \equiv \left(\frac{-r\tan(\theta_{T1})}{\cos(\phi_{T1})}, \frac{r\tan(\phi_{T1})}{\cos(\theta_{T1})}, r \right) \tag{5.27}$$

$$P_{R2} \equiv \left(\frac{-r\tan(\theta_{T2})}{\cos(\phi_{T2})}, \frac{r\tan(\phi_{T2})}{\cos(\theta_{T2})}, r \right) \tag{5.28}$$

Their respective distances from $P_O \equiv (0, 0, r)$ are:

$$P_{R1}P_O = \left[\left(\frac{r\tan(\theta_{T1})}{\cos(\phi_{T1})} \right)^2 + \left(\frac{r\tan(\phi_{T1})}{\cos(\theta_{T1})} \right)^2 \right]^{\frac{1}{2}} \tag{5.29}$$

$$P_{R2}P_O = \left[\left(\frac{r\tan(\theta_{T2})}{\cos(\phi_{T2})} \right)^2 + \left(\frac{r\tan(\phi_{T2})}{\cos(\theta_{T2})} \right)^2 \right]^{\frac{1}{2}} \tag{5.30}$$

Now, if the screen is moved by a distance Δr such that $r + \Delta r = r_1$, then the two spot positions P'_{R1}, P'_{R2} will be given by (Fig. 5.7):

$$P'_{R1} \equiv \left(\frac{-r_1\tan(\theta_{T1})}{\cos(\phi_{T1})}, \frac{r_1\tan(\phi_{T1})}{\cos(\theta_{T1})}, r_1 \right) \tag{5.31}$$

$$P'_{R2} \equiv \left(\frac{-r_1\tan(\theta_{T2})}{\cos(\phi_{T2})}, \frac{r_1\tan(\phi_{T2})}{\cos(\theta_{T2})}, r_1 \right) \tag{5.32}$$

Their respective distances from $P'_O \equiv (0, 0, r_1)$ are:

$$P'_{R1}P'_O = \left[\left(\frac{r_1\tan(\theta_{T1})}{\cos(\phi_{T1})} \right)^2 + \left(\frac{r_1\tan(\phi_{T1})}{\cos(\theta_{T1})} \right)^2 \right]^{\frac{1}{2}} \tag{5.33}$$

$$P'_{R2}P'_O = \left[\left(\frac{r_1\tan(\theta_{T2})}{\cos(\phi_{T2})} \right)^2 + \left(\frac{r_1\tan(\phi_{T2})}{\cos(\theta_{T2})} \right)^2 \right]^{\frac{1}{2}} \tag{5.34}$$

The scale change in the first case, m_1, is defined as:

$$\begin{aligned} m_1 =& \frac{P'_{R1}P'_O}{P_{R1}P_O} \\ =& \frac{r_1}{r} \end{aligned} \tag{5.35}$$

Similarly, for the second case, scale change, m_2, is defined as:

$$\begin{aligned} m_2 =& \frac{P'_{R2}P'_O}{P_{R2}P_O} \\ =& \frac{r_1}{r} \end{aligned} \tag{5.36}$$

It is evident from the above two equations that the scale change (m) is spot position independent and only depends on the object distance from the centre O of the GCS, i.e.

$$m = m_1 = m_2 = \frac{r_1}{r} \tag{5.37}$$

Now, for a thin-lens, we know:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{5.38}$$

If the object distance (u) is large, we can approximate the above equation as:

$$\begin{aligned} \frac{1}{f} =& \frac{1}{v} \\ \implies v \approx& f \end{aligned} \tag{5.39}$$

Magnification of a thin-lens (m_l) is defined as:

$$m_l = -\frac{v}{u} = -\frac{f}{u} \quad \text{[from equation 5.39]} \tag{5.40}$$

Considering $u = r_1$ in our case, we have magnification due the lens as:

$$m_l = -\frac{f}{r_1} \tag{5.41}$$

From the equation (5.41) we can conclude that any linear magnification of an object dimension affected by the movement of the object will be nullified by the magnification factor of the lens.

This, in turn, means that if the principal point of the camera coincided with the centre of the GCS, there would not have been any perceivable change in the spot positions on the image plane with change of r_1. The model thus highlights the effects

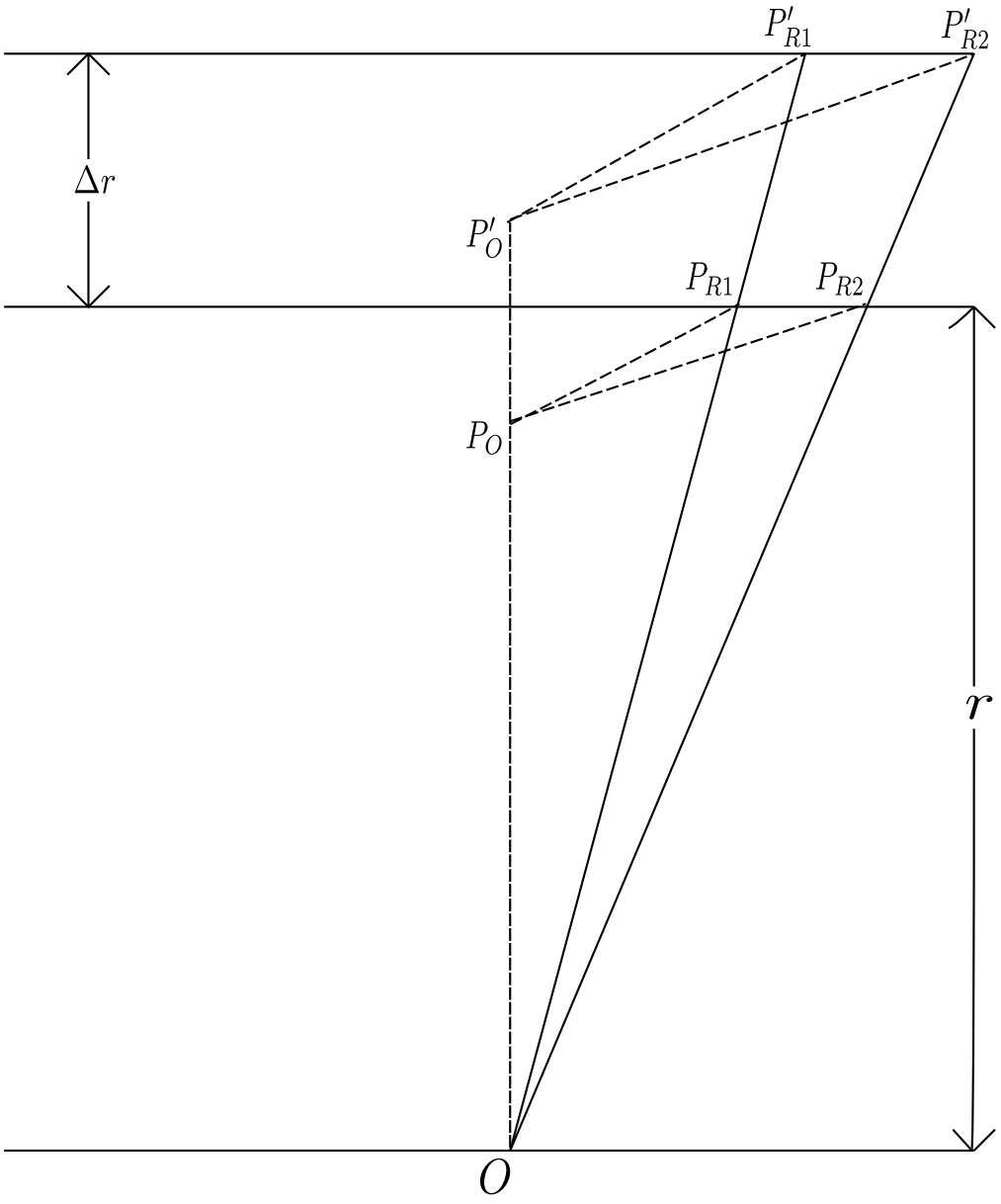


Figure 5.7: Geometry showing all object points of the projected pattern are linearly magnified with change of depth.

of the translational shifts of the principal point of the lens with respect to O, the centre of the GCS. It is now apparent that a shift in either the x or y direction of the camera only will make the spots on the image plane move in the same direction as that of the shift, provided there is no z-displacement. This, in other words, means that if the principal point of the camera lens is given a shift in any only one direction (lateral or vertical), excluding the axial one, then the task of spot tracking will be straightforward.

5.4.6 Use of the model

In general, the model developed can be used in the following ways:
(1) to estimate depth by either determining the x-coordinates or the y-coordinates of the generated spots (refer to equations (5.17) and (5.18)) on the image plane. It will depend upon the situation whether estimating the x-coordinates of the spots will be considered easier than estimating the y-coordinates of the spots or vice versa;

(2) equations of the model shows how each of the spots will move on the image plane with change of r. Such *a-priori* knowledge of the path a particular spot will follow on the image plane facilitates spot-tracking and hence determination of object depth;

(3) the spot movement guiding factor equation (equation 5.25) can be used effectively to increase the working volume of the system without explicitly encoding the generated spots; how this can be achieved is described in Section 5.6.

5.4.7 Non-inverted equations

Bearing in mind that the non-inverted version of the model might be considered more tractable, a revised form of the equations (5.15), (5.16), and (5.25) are presented below. The equations have been been derived assuming that the thin lens is placed a focal length (f) distance behind the image plane. Thus, the co-ordinates of the centre of the image plane, I, will be $(\varphi_x, \varphi_y, \varphi_z + f)$.

Noting in this case $z_{PI} = \varphi_z + f$ we get from equation (5.13):

$$\left[\frac{x_{PI} - \varphi_x}{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}\right] = \left[\frac{y_{PI} - \varphi_y}{\frac{r\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y}\right] = \left[\frac{f + \varphi_z - \varphi_z}{r - \varphi_z}\right] = \left[\frac{f}{r - \varphi_z}\right] \tag{5.42}$$

Therefore, the non-inverted forms of the equations (5.15), (5.16) are:

$$x_{PI}\mid_{non-inverted} = \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z}\right](f) + \varphi_x \tag{5.43}$$

$$y_{PI}\mid_{non-inverted} = \left[\frac{\frac{r\tan(\phi_T)}{\cos(\theta_T)} - \varphi_y}{r - \varphi_z}\right](f) + \varphi_y \tag{5.44}$$

Applying the same assumption, we get the gradient of the path of a first quadrant spot on the image plane as:

$$\frac{dy_{PI}}{dx_{PI}} = \left[\frac{-\varphi_z\frac{\tan(\phi_T)}{\cos(\theta_T)} + \varphi_y}{\varphi_z\frac{\tan(\theta_T)}{\cos(\phi_T)} + \varphi_x}\right] \tag{5.45}$$

5.5 Model parameter estimation

To meet the objectives of developing the model, the following parameters have to be *a priori* estimated as accurately as possible:
(1)the focal length of the lens of the camera;
(2)the horizontal scan angle (θ), and the vertical scan angle (ϕ);
(3)the x-shift and/or the y-shift of the principal point of the lens with respect to the centre of the GCS; and
(4)the z-shift of the principal point of the lens with respect to the centre of the GCS.

5.5.1 Estimating the focal length of the lens of the monochrome camera

To determine the focal length of the lens, f, of the camera, to locate the position of its principal point and to rectify the distortions in the images captured by the camera, the camera lens was calibrated using Zhang's method [41], a straightforward camera calibration technique. From the camera calibration results tabulated in Table 5.1, we get the focal length of the lens as: 898.38±7.46 pixels. Since the monochrome camera used comes fitted with a $\frac{1}{3}$" CCD sensor with a pixel pitch specification of 4.65μm,

the focal length of the lens can also be expressed as:

$$\begin{aligned} f &= [898.38 \pm 7.46] \times 4.65 \times 10^{-6}\ m \\ &= [4.18 \pm 0.035]\ mm \end{aligned} \tag{5.46}$$

Note that a full calibration procedure, mainly to determine the focal length of the lens and to correct the images viewed through it, is necessary as the entire modeling is done based on thin lens theory.

Table 5.1: Camera calibration results

Parameter	Best estimate	Error
Focal length	898.38 pixel	$\pm$7.46 pixel
Principal point	[533.80 371.81]	$\pm$[7.0 7.88]

5.5.2 Estimating the horizontal scan angle, θ, and the vertical scan angle, ϕ

This subsection describes the steps taken to measure the two intrinsic parameters of the developed model: the horizontal scan angle θ, and the vertical scan angle ϕ. It will also become clear in this subsection, how through the use of the same steps, the location of the DOE plane was estimated. This was deemed necessary as the laser-DOE arrangement came assembled within a case with a lens-system fixed at its front. First, the structured pattern was projected on a flat foam-board placed roughly $0.3\ m\ -0.4\ m$ away from the LDA arrangement. The distance, h_1, between the central spot and one adjacent to it was measured using a steel ruler with a resolution of $0.5\ mm$. The distance, h_1, came out as:

$$h_1 = [4 \pm 0.5]\ mm \tag{5.47}$$

The flat foam board was then shifted by a certain distance, d, from its intial position and again the distance, h_2, between the same two spots was measured. Both the distances d and h_2 were measured using the same steel ruler. The following are the

estimates of the measured distances:

$$d = \left[1 \pm 0.5 \times 10^{-3}\right] \; m \tag{5.48}$$

$$h_2 = [16 \pm 0.5] \; mm \tag{5.49}$$

Now from the geometry shown in Fig. 5.8, the angle, ϕ, was measured as:

$$\phi = \frac{(h_2 - h_1)}{d} \tag{5.50}$$

The best estimate of ϕ was obtained using the best estimates of h_1, h_2 and d, as follows:

$$\begin{aligned} \phi =& \frac{(16-4) \times 10^{-3}}{1} \; radians \\ =& 0.0120 \; radians \end{aligned} \tag{5.51}$$

Now, representing the errors on h_1, h_2 and d as σ_{h_1}, σ_{h_2} and σ_d, respectively, the bounds of the error, σ_ϕ, on the measurement of ϕ, was obtained as [90], [91]:

$$\sigma_\phi = \sqrt{(\frac{\delta\phi}{\delta h_1}\sigma_{h_1})^2 + (\frac{\delta\phi}{\delta h_2}\sigma_{h_2})^2 + (\frac{\delta\phi}{\delta d}\sigma_d)^2} \tag{5.52}$$

Expanding and putting the individual parameter values we get σ_ϕ as:

$$\begin{aligned} \sigma_\phi =& \sqrt{(0.5 \times 10^{-3})^2 + (0.5 \times 10^{-3})^2 + (12.0 \times 0.5 \times 10^{-6})^2} \\ \approx& 0.5 \times 10^{-3} \; radians \end{aligned} \tag{5.53}$$

Therefore,

$$\begin{aligned} \phi =& \left[0.012 \pm 0.5 \times 10^{-3}\right] \; radians \\ =& [0.69 \pm 0.03] \; deg \end{aligned} \tag{5.54}$$

Note that in all subsequent experiments, the model parameter, ϕ, was taken as $0.69°$. The location of the DOE plane (the distance d_{DOE}) was then calculated and marked using the following relationship:

$$d_{DOE} = \frac{h_2}{\tan\phi} = \frac{h_2}{\frac{(h_2 - h_1)}{d}} = \frac{dh_2}{(h_2 - h_1)} \tag{5.55}$$

Upon substituting the individual parameter values this is found to be $1.33 \; m$.

The error in the estimate, $\sigma_{d_{DOE}}$, was calculated as follows:

$$\sigma_{d_{DOE}} = \sqrt{(\frac{\delta d_{DOE}}{\delta x_1}\sigma_{x_1})^2 + (\frac{\delta d_{DOE}}{\delta x_2}\sigma_{x_2})^2 + (\frac{\delta d_{DOE}}{\delta d}\sigma_d)^2} \tag{5.56}$$

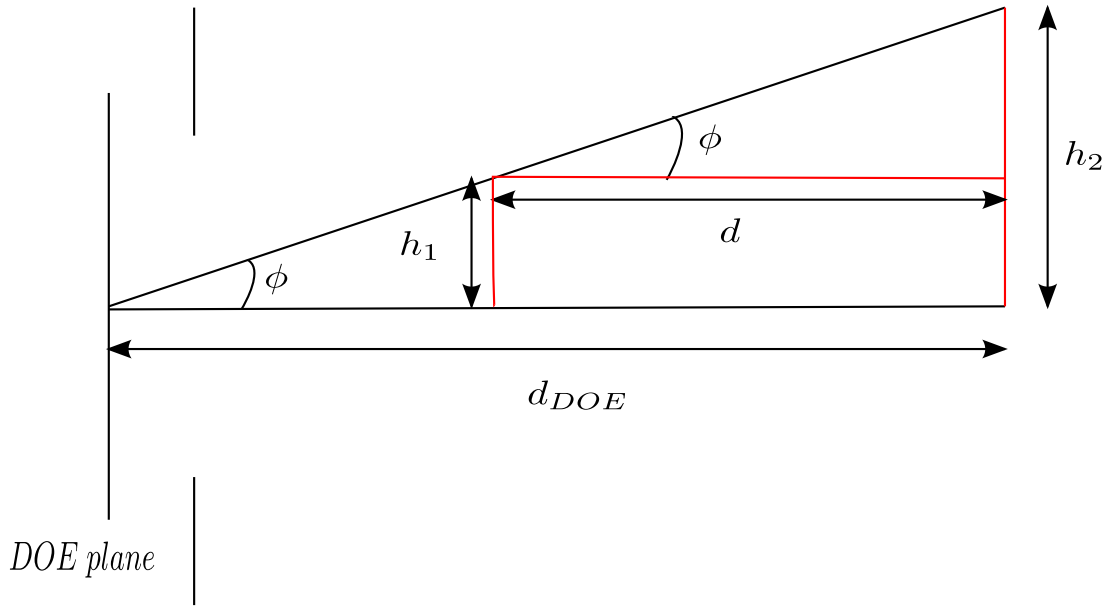


Figure 5.8: Geometry used to estimate the vertical scan angle, ϕ, and to mark the location of the DOE plane.

Expanding and substituting the individual parameter values gives a value of $\sigma_{d_{DOE}}$ equal to $3.28 \times 10^{-3}\ m$.
A similar procedure was adopted to estimate the horizontal scan angle, θ, and it was noted that 0.69° can also be used as a good estimate of the parameter.

5.5.3 Estimating φ_x and φ_z

From the description of the developed model it becomes apparent that to determine the depth of an object point the x-shift (or the y-shift) and the z-shift of the principal point of the lens (the centre, C, of the co-ordinate system fixed on the lens) have to be determined with respect to the GCS fixed on the centre of the DOE. To accomplish this task, the central spot of the pattern projected on the foam board was tracked using an iris fitted to a mount whose height can be adjusted and by giving the board regular shifts away from the LDA (Fig. 5.9). The foam board was initially kept at a distance of 0.5 m from the DOE plane and the co-ordinates of the central spot on the image plane estimated using a standard centroid finding algorithm. The foam board was then given a shift of 0.075 m and the procedure repeated. In this way the co-ordinates of the central spot were determined at regular intervals until the foam board was 1.550 m away from the DOE plane. The estimated x-coordinates

(x_{PIobs}) and y-coordinates (y_{PIobs}) of the centroid of the central spot for every shift has been tabulated in Table 5.2.

Now, the model equation (5.15) gives the x-coordinate of an object point on the image plane as:

$$x_{PI} = \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z}\right](-f) + \varphi_x$$

Since the central spot was tracked, $\theta_T = \phi_T = 0$. Therefore,

$$x_{PI} = \left[\frac{\varphi_x f}{r - \varphi_z}\right] + \varphi_x \tag{5.57}$$

The left-hand side of equation (5.57) can be expanded as:

$$x_{PI} = \varphi_x + (x_{PIobs} - \varsigma)\,\rho \tag{5.58}$$

where, in equation (5.58), ς is the horizontal pixel index corresponding to φ_x and ρ is the pixel pitch of the $\frac{1}{3}$" CCD sensor.

Combining the two equations (5.57) and (5.58) we get:

$$\varphi_x + ((x_{PIobs} - \varsigma)\,\rho = \left[\frac{\varphi_x f}{r - \varphi_z}\right] + \varphi_x \tag{5.59}$$

Therefore,

$$x_{PIobs} = \varsigma + \frac{\varphi_x f}{r\rho}\left(1 - \frac{\varphi_z}{r}\right)^{-1} \tag{5.60}$$

Expanding $(1 - \frac{\varphi_z}{r})^{-1}$ and neglecting the higher order terms [$\because \varphi_z \ll r$] we have:

$$x_{PIobs} = \varsigma + \frac{\varphi_x f}{r\rho}\left(1 + \frac{\varphi_z}{r}\right) \tag{5.61}$$

$$= \varsigma + \beta_k\left(\frac{\varphi_x f}{\rho}\right) + \beta_k^2\left(\frac{\varphi_x \varphi_z f}{\rho}\right) \tag{5.62}$$

$$= k_0\beta_k^0 + k_1\beta_k^1 + k_2\beta_k^2 \tag{5.63}$$

where, in equation (5.63), $\beta_k = \dfrac{1}{r}$, $k_0 = \varsigma$, $k_1 = (\frac{\varphi_x f}{\rho})$ and $k_2 = (\frac{\varphi_x \varphi_z f}{\rho})$. Fifteen values of the location of the central spot on the image plane were noted against the corresponding values of β_k (Table 5.2). Best estimates of k_0, k_1 and k_2 in the least squared error sense [92] were then determined. The following are the values of k_0, k_1 and k_2 determined through the application of the method:

$$\begin{aligned} k_0 &= \varsigma = 521.76 \\ k_1 &= -\,66.42 \\ k_2 &= -\,3.29 \end{aligned} \tag{5.64}$$

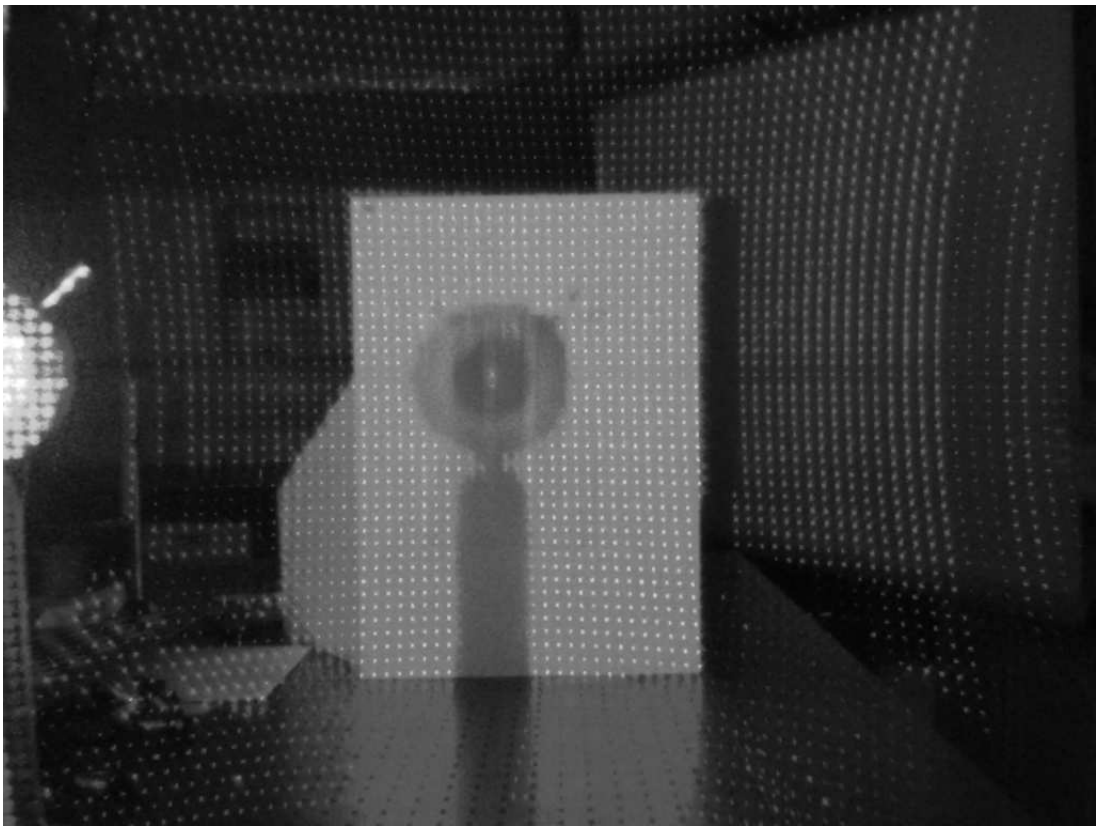


Figure 5.9: The central spot being tracked using an iris fitted to a mount whose height can be adjusted. (The shadow of the iris fitted to the mount can be seen on the projection screen.)

The norm of the residuals came out as 2.26. This uncertainty can solely be assumed as the error incurred in determining k_0, as the other factors are weighted by $\frac{1}{r}$ and $\frac{1}{r^2}$, respectively, making their contributions to the total error negligible.
Through the use of the least square method, we got the horizontal pixel index of the central pixel as as 521.76 ± 2.26. Also, the camera calibration results indicate the horizontal pixel index of the principal point of the lens as 533.80 ± 7.0. These deviations often creep in due to misalignments during the camera parts assembly process. Moreover, it has to be always borne in mind that a thin lens is assumed while modeling the arrangement, instead of the lens system fitted to the camera.
Using the value of k_1, we got the x-shift of C with respect to the GCS as:

$$\varphi_x = \frac{k_1 \times \rho}{f} \tag{5.65}$$

Putting the individual parameter values, we obtain φ_x as:

$$\begin{aligned}\varphi_x &= \left[\frac{-66.42 \times 4.65 \times 10^{-6}}{4.18 \times 10^{-3}}\right] \\ &= -73.89 \times 10^{-3}\ m\end{aligned}$$

Similarly, using the value of k_2, we got the z-shift of C with respect to the GCS as:

$$\varphi_z = \frac{k_2 \times \rho}{\varphi_x \times f} \tag{5.66}$$

Putting the individual parameter values, we obtain φ_z as:

$$\begin{aligned}\varphi_z &= \left[\frac{-3.29 \times 4.65 \times 10^{-6}}{-73.89 \times 10^{-3} \times 4.18 \times 10^{-3}}\right] \\ &= 0.049\ m\end{aligned}$$

Note that these values for the x-shift and the z-shift of C with respect to the GCS only indicate the estimates that best represent the data in a least squared error sense.

5.5.4 Testing the model

Equipped with all the required model parameter values we have proceeded to test the performance of the model. Several other spots have been tracked using the method described earlier and for each position the co-ordinates of the centroid of the spot

on the image plane noted. Four such series of observations made are tabulated in Tables 5.2, 5.3, 5.4 and 5.5. Depth has been calculated using the x-coordinates of the centroid (x_{PIobs}) using equation (5.17). The expanded form of the equation, used for depth calculation is:

$$\begin{aligned} r &= \left[\frac{f\varphi_x + (x_{PI} - \varphi_x)\,\varphi_z}{(x_{PI} - \varphi_x) - \frac{f\tan(\theta_T)}{\cos(\phi_T)}} \right] \\ &= \left[\frac{f\varphi_x + [\{\varphi_x + (x_{PIobs} - \varsigma)\rho\} - \varphi_x]\,\varphi_z}{[\{\varphi_x + (x_{PIobs} - \varsigma)\rho\} - \varphi_x] - \frac{f\tan(\theta_T)}{\cos(\phi_T)}} \right] \text{ [Using equation (5.58)]} \\ &= \left[\frac{f\varphi_x + (x_{PIobs} - \varsigma)\rho\varphi_z}{(x_{PIobs} - \varsigma)\rho - f\frac{\tan\theta_T}{\cos\phi_T}} \right] \end{aligned} \tag{5.67}$$

The predicted depth estimates, and the corresponding signed errors, calculated with

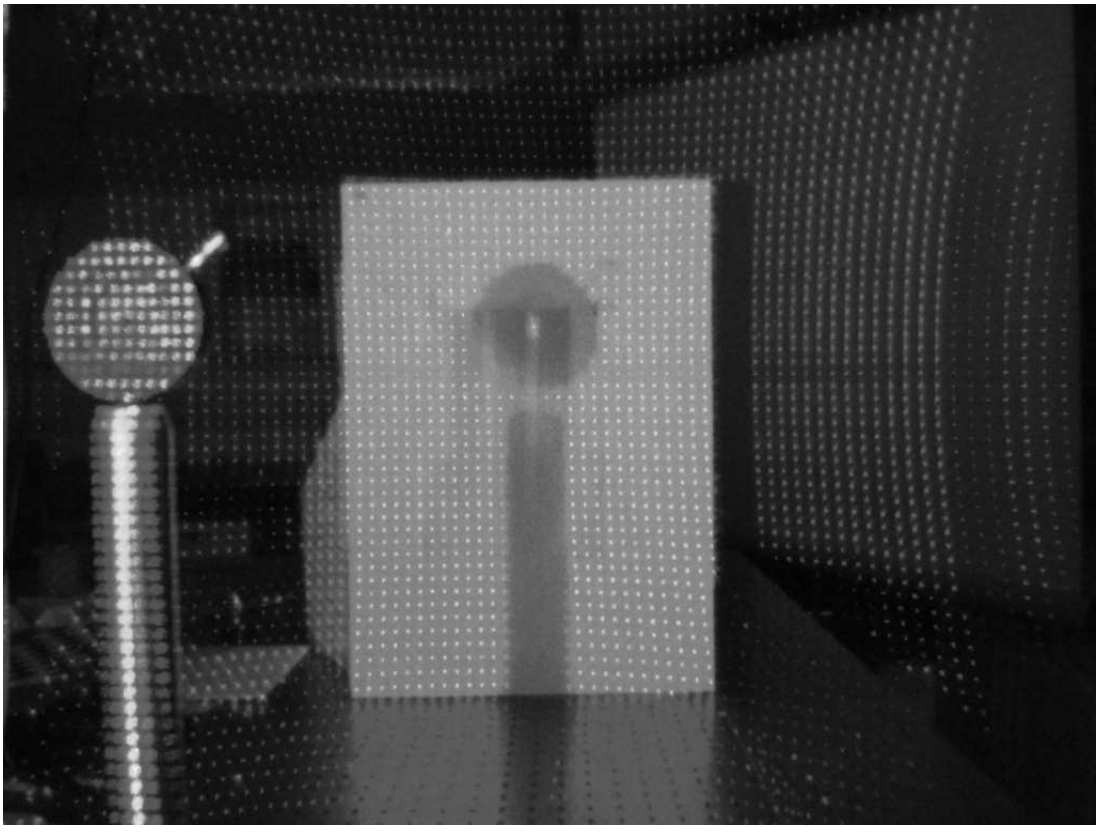


Figure 5.10: Spot $\langle 1, :, 4\theta, 4\phi \rangle$ being tracked to assess the performance of the developed model.

respect to the ground truth data, have been included in the individual tables.

Table 5.2: Spot tracked: $\langle 1, :, 0\theta, 0\phi \rangle$

Distance, r, from the DOE plane in mm $\pm$ 0.81 mm	$\beta_k = \frac{1}{r}$ (in m^{-1})	x_{PIobs}	y_{PIobs}	Predicted distance, r^{Θ}, in m	Signed error $\left[\frac{(r-r^{\Theta})}{r}\right]$ (%)
500	2.00	376.1765	349.9955	0.505	-1.00
575	1.74	395.7681	348.9929	0.576	-0.17
650	1.54	411.8265	349.0146	0.653	-0.46
725	1.38	423.8461	349.9792	0.727	-0.28
800	1.25	432.8622	349.9611	0.796	$+0.50$
875	1.14	440.8643	348.9920	0.870	$+0.57$
950	1.05	448.8252	349.9656	0.960	-1.05
1025	0.98	454.8480	349.9656	1.042	-1.66
1100	0.91	459.8474	348.0129	1.122	-2.00
1175	0.85	462.7908	348.9518	1.175	0.00
1250	0.80	465.7406	348.9258	1.235	$+1.20$
1325	0.75	469.7646	349.9204	1.326	-0.07
1400	0.71	472.7033	348.9120	1.403	$+0.21$
1475	0.68	475.0883	350.4444	1.472	$+0.20$
1550	0.64	477.1063	348.8412	1.536	$+0.90$

Table 5.3: Spot tracked: $\langle 1, :, 4\theta, 4\phi \rangle$

Distance, r, from the DOE plane in mm $\pm$ 0.81 mm	x_{PIobs}	y_{PIobs}	Predicted distance, r^{Θ}, in m	Signed error $\left[\frac{(r-r^{\Theta})}{r}\right]$ (%)
450	418.7678	303.8780	0.488	−8.40
525	439.6673	304.9383	0.561	−6.80
600	455.7283	304.9218	0.637	−6.20
675	466.7538	304.9150	0.702	−4.00
750	476.9097	304.9481	0.778	−3.7
825	484.9679	305.9360	0.850	−3.00
900	491.9522	305.9958	0.927	−3.00
975	496.9178	304.9460	0.991	−1.60
1050	500.7332	305.9472	1.047	+0.28
1125	506.7801	305.9745	1.151	−2.20
1200	508.6818	306.9961	1.188	+1.60
1275	512.6040	305.9873	1.272	+0.24
1350	515.3813	305.7844	1.341	+0.66
1425	517.6548	305.8199	1.403	+1.50
1500	519.5950	307.6852	1.460	+2.60

Table 5.4: Spot tracked: $\langle 1, :, 15\theta, 10\phi \rangle$

Distance, r, from the DOE plane in mm $\pm$ 0.81 mm	x_{PIobs}	y_{PIobs}	Predicted distance, r^{Θ}, in m	Signed error $\left[\frac{(r-r^{\Theta})}{r}\right]$ (%)
450	547.7672	231.8551	0.468	−3.78
525	566.8738	231.8987	0.534	−1.71
600	581.8096	233.8524	0.603	−0.50
675	594.9316	232.9167	0.682	−4.00
750	602.8827	233.9934	0.741	+1.20
825	607.9270	233.9607	0.785	+4.85
900	614.9279	233.9901	0.857	+4.78
975	620.8628	234.9530	0.929	+4.72
1050	625.9080	235.9720	1.002	+4.57
1125	629.8471	235.9634	1.067	+5.16
1200	632.8118	235.9375	1.123	+6.42
1275	635.7765	235.9213	1.185	+7.06
1350	639.7722	235.9686	1.280	+5.18
1425	641.7461	236.9007	1.334	+6.39
1500	643.3399	235.3399	1.381	+7.93

Table 5.5: Spot tracked: $\langle 1, :, 20\theta, 5\phi \rangle$

Distance, r, from the DOE plane in mm ± 0.81 mm	x_{PIobs}	y_{PIobs}	Predicted distance, r^{Θ}, in m	Signed error $\left[\frac{(r-r^{\Theta})}{r}\right]$ (%)
450	612.6597	288.9118	0.476	−5.78
525	629.8501	290.9374	0.540	−2.86
600	644.8549	290.9695	0.616	−2.67
675	654.9205	289.9730	0.687	−1.78
750	663.9411	291.9848	0.752	−0.27
825	669.8933	290.9424	0.810	+1.82
900	676.8957	292.9625	0.890	+1.11
975	681.8153	292.9986	0.958	+1.74
1050	684.7980	292.0434	1.005	+4.29
1125	688.7730	292.9432	1.073	+4.44
1200	692.6385	292.0288	1.154	+3.83
1275	695.7584	292.9303	1.227	+3.76
1350	698.8208	292.8961	1.308	+3.11
1425	701.3124	292.7324	1.384	+2.88
1500	702.5244	291.5244	1.424	+5.07

From the data listed in the tables, it can be inferred that the developed model satisfactorily estimates depths of object points that lie in the central region of the projected pattern. In fact, the figures suggest that if the object point range can be constrained between 0.7 m to 1.4 m, then the error incurred in prediction can be limited to 5%. The data also reflects the fact that the error in prediction increases when elements lying outside the central region, particularly when $\phi_T > \pm 10\phi$ is used to estimate depth. This, in turn, suggests that though the lens system has been calibrated, radial distortions have not been fully compensated for.

Note that in this thesis, no attempt has been made to estimate the y-shift of the principal point of the lens, C, with respect to the GCS. This is because the major objective of this thesis is to perform scene segmentation using depth or depth based estimates. This can be adequately achieved by just estimating the focal length of the lens, the x-shift and z-shift of the principal point with respect to the GCS and by noting the pixel pitch of the sensor used from a specification book (refer to model equations (5.15) and (5.17)). Determining the y-shift of C with respect to the GCS is not considered redundant when generating full world-coordinates of object points as may be required in applications like multi-camera based scene understanding and object tracking. The task of measuring the y-shift may also be undertaken to alleviate some of the difficulties associated with object point tracking on the camera image plane. However, this task can also be accomplished using other techniques, as will be explained in Chapter 6. It should also be noted that the data obtained (y_{PIobs} from all the tables) suggests that there is either no or negligible y-shift of the camera lens' principal point with respect to the GCS. Even if there is a very small shift, it is not hard to understand that it is well buried in measurement error noise.

It is also evident from the process described, that determining the horizontal pixel index, ς, of the sensor, corresponding to φ_x, is one of the processes susceptible to error. However, this task can be circumvented by measuring the co-ordinates of the projections on the image plane of an object point kept at two different distances from O, one of those being known *a priori*. A theoretical description of how this can be done is detailed in the next subsection.

5.5.5 Estimating depth from two measurements

Depth of an object point can also be estimated by noting the change of locations of its projections on the image plane corresponding to its two different positions. Of these two different positions, the distance of one from the centre of the GCS has to be known *a priori* and may be fixed. Here also, calculating either the x-coordinates or the y-coordinates of the projections is considered adequate to accomplish the task of depth estimation. The process is detailed as follows:
Let $(\psi_x, \psi_y, \varphi_z - f)$ be an arbitrary point on the image plane and ς_{ψ_x} be the horizontal

pixel index corresponding to ψ_x. Now for an arbitrary object point, kept at a distance of r meters away from the centre of the GCS (considered to be known *a priori*), let the co-ordinates of its projection on the image plane be (x_{PIobs1}, y_{PIobs1}). Using model equations (5.15) and (5.58), we can write this as:

$$\psi_x + ((x_{PIobs1} - \varsigma_{\psi_x})\,\rho = \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z}\right](-f) + \varphi_x \tag{5.68}$$

Now, let us suppose that the object point is shifted and it is r_1 meters away from the centre of the GCS. Let the corresponding co-ordinates of its projection on the image plane be (x_{PIobs2}, y_{PIobs2}). The task is to find out the changed depth, r_1, of the same object point. Again using model equations (5.15) and (5.58) we have:

$$\psi_x + ((x_{PIobs2} - \varsigma_{\psi_x})\,\rho = \left[\frac{\frac{-r_1\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r_1 - \varphi_z}\right](-f) + \varphi_x \tag{5.69}$$

Subtracting equation (5.68) from equation (5.69) we get:

$$(x_{PIobs2} - x_{PIobs1})\rho = \left[\frac{\frac{-r_1\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r_1 - \varphi_z}\right](-f) + \varphi_x - \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z}\right](-f) - \varphi_x$$

or,

$$(x_{PIobs2} - x_{PIobs1})\rho + \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z}\right](-f) = \left[\frac{\frac{-r_1\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r_1 - \varphi_z}\right](-f) \tag{5.70}$$

Substituting χ for $(x_{PIobs2} - x_{PIobs1})\rho + \left[\frac{\frac{-r\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r - \varphi_z}\right](-f)$, equation (5.70) can rewritten as:

$$\chi = \left[\frac{\frac{-r_1\tan(\theta_T)}{\cos(\phi_T)} - \varphi_x}{r_1 - \varphi_z}\right](-f) \tag{5.71}$$

Rearranging equation (5.71), we get r_1 as:

$$r_1 = \left[\frac{\varphi_x f + \chi\varphi_z}{\chi - (\frac{\tan\theta_T}{\cos\phi_T})f}\right] \tag{5.72}$$

The difficulties associated with finding out the horizontal pixel index corresponding to the x-shift of C with respect to the GCS, φ_x, which is not needed in the method described above, clearly underpins its usefulness.

A section of the spot tracking data is taken from Table 5.4 where the error incurred in predicting the range varied between 5.16% to 7.06% over a range of 1.125 m

- 1.425 m, the foam board being shifted by regular steps of 0.075 m. It has to remembered that lens distortion that has not been fully compensated for, mainly causes such errors in range estimation. Data from this section of the table has been used to assess the performance of the modified depth perception method based on difference of co-ordinate. The calculated x-coordinate of the projection of the object point ($\langle 1, :, 15\theta, 10\phi \rangle$), with the foam board been kept at a distance of 1.5 m away from the DOE plane, is taken as the value for x_{PIobs1}. Therefore, $x_{PIobs1} = 643.3399$. The distance of 1.50 m is taken as the value for r, which has to be known *a priori*.

Now for every successive position of the object point (from 1.125 m to 1.425 m at regular steps of 0.075 m), the x-coordinate of its projection on the image plane is noted in terms of pixel index value, and equation (5.72) applied to estimate depth. The predicted depth values using the modified method are listed in Table 5.6. Signed errors have also been measured by comparing the predicted range estimates with the ground truth data. The results obtained show that the corresponding error estimates now vary between $0.16\% - 1.40\%$, as against $5.16\% - 7.06\%$ when the original method was used, thus validating the usefulness of the modified method of depth perception.

Table 5.6: Depth sensing using difference of co-ordinate method

Distance, r, from the DOE plane in mm $\pm$ 0.8 mm	x_{PIobs}	Predicted distance, r^{ψ}, in m	Signed error $\left[\frac{(r-r^{\psi})}{r}\right]$ (%)
1.125	688.7730	1.135	-0.89
1.200	692.6385	1.198	$+0.16$
1.275	695.7584	1.270	$+0.39$
1.350	698.8208	1.382	-2.29
1.425	701.3124	1.445	-1.40

5.6 Increasing the working volume of the system

In this section a method is proposed that can be utilised to increase the operational volume of the active depth sensing system without explicitly encoding the structured coded pattern. Recall from Section 5.4.4, that if there is no z-shift of the principal point of the lens with respect to the GCS, then the gradient of the path the projection of an arbitrary object point will follow on the image plane solely depends upon φ_x and φ_y (refer to equation (5.26) of the Section). Also we use the fact that if the the principal point of the lens is given equal shifts in both x and y directions, z-shift being 0, then all the spots on the image plane will follow a straight line path of slope, $\tan 45^\circ = 1$. Ambiguity arises when the projection of any arbitrary spot moves across the location of the projection of another object point lying in close proximity of the spot being considered.

The distance, d_p, between the projection of the central spot and that of an adjacent spot (lying on the line: $y = 0,\ z = r$ on the foam-board) on the image plane is given by:

$$\begin{aligned} d_p &= \left[\frac{-r\tan(\theta) - \varphi_x}{r - \varphi_z}\right](-f) - \frac{\varphi_x f}{r - \varphi_z} \\ &= \left[\frac{fr\tan\theta}{r - \varphi_z}\right] \end{aligned} \tag{5.73}$$

A hypothetical square of sides equal to $\left[\frac{2fr\tan\theta}{r-\varphi_z}\right]$ is now drawn around the projection of the central spot on the image plane. If $\varphi_z = \varphi_y = 0$, then the projections of all the spots on the image plane will move along the x- axis; for the projection of the central spot, ambiguity will creep in as soon as it moves by a distance of $\left[\frac{fr\tan\theta}{r-\varphi_z}\right]$. However, if $\varphi_z = 0$ and $\varphi_x = \varphi_y$, then the projection of the central spot will move along the diagonal of the square and thus it can travel further by a distance Δ_{d_p} before ambiguity creeps in. Δ_{d_p} is given by:

$$\begin{aligned} \Delta_{d_p} &= \left[\sqrt{2}\frac{fr\tan\theta}{r - \varphi_z}\right] - \left[\frac{fr\tan\theta}{r - \varphi_z}\right] \\ &\approx 0.41\left[\frac{fr\tan\theta}{r - \varphi_z}\right] \end{aligned} \tag{5.74}$$

Thus, by giving the principal point of the lens deliberate shifts along the x and y direction with respect to the GCS, the working volume of the system can be approximately increased by 40%.

Finally, it should be noted that more robust and efficient camera calibration techniques need to be investigated to utilise the full range of the projected pattern and to ensure that the predicted depths of object points, situated at a specific distance from the centre of the GCS, are approximately the same. However, to perform the task of scene segmentation, explicit depth maps may not be needed. Instead, a foreground scene can be segmented by simply sensing the projected structured pattern disparity. A method to accomplish this task has been developed in Chapter 6.

5.7 Summary

In this chapter an active structured light based depth sensing system has been modeled to facilitate its understanding and use. Through the use of the model equations depth of an object point can be estimated either by estimating the x-coordinate or the y-coordinate of its projection on the image plane of a sensor used to view the scene. How the model parameters can be *a priori* estimated has been discussed in depth in this chapter and the functioning of the depth sensing arrangement demonstrated. A modified method to perceive depth through the determination of coordinates of the projections of an object point situated at two different locations has also been presented.

Chapter 1 of the thesis discusses the disadvantages of explicitly encoding the projected structured pattern. A theory has been proposed in this chapter that can be used to increase the operational volume of the described structured light based depth sensing system by approximately 40% without deploying an explicit pattern coding scheme.

It has been noted in this chapter that for an efficient use of the developed model, more complex camera calibration techniques need to be investigated and their individual performances assessed. However, to perform the task of scene segmentation, determination of only structured pattern disparity may be considered sufficient thus making redundant the requirement of generation of explicit depth maps. A method to accomplish the task of foreground scene extraction by solely noting the structured pattern spot position disparity has been developed in the next chapter.

Chapter 6

DENOISING VIDEO FRAMES CONTAINING REGIONS SEGMENTED USING STRUCTURED SPOT POSITION DISPARITY ESTIMATES

6.1 Introduction

A practical mathematical model of an active range camera has been developed in the previous chapter. However, it should be mentioned that, in general, range estimates (measurements) from a range-camera are used to determine the world-coordinates of a point on an object or to do robust realistic 3−D modeling. This, in turn, means that to segment multiple partially overlapping or non-overlapping foreground objects or a single object from the background scene, explicit depth maps need not be generated using the model. Instead, in such cases deformation of the projected structured pattern (spot shifts) can be efficiently used to meet the objective of foreground scene segmentation. In this chapter a method has been developed to identify those spots on the camera image plane whose positions get shifted from their initial

locations due to an object coming in-between the camera and the screen on which the structured pattern is originally projected.
However, it has also been noted that frames, containing foreground objects segmented using the developed method, come out contaminated with, generally isolated, noisy pixel clusters (blocks). In addition to identifying spots with shifted locations, two different median filtering schemes have been developed in this chapter to remove those noisy blocks from the output frames. An analogy of one of the median filtering schemes with conventional discrete morphological operators has also been included in this chapter.

6.2 Chapter organisation

The chapter is organised as follows: Section 6.3 describes the experimental set-up used to generate the structured spot position disparity estimates. A custom-made median filter and a metric to assess its performance are developed in Section 6.4; the same section also reports the results obtained after applying the developed filtering scheme to denoise frames contaminated with noisy pixel clusters. Section 6.5 elaborates another scheme that uses multistage max/median and min/median filters to remove the noise pixel blocks from the noise corrupted frames. Results obtained after applying the filtering scheme on block-noise corrupted frames and an analogy of multistage max/median and min/median filters with basic discrete morphological operators have also been included in the same section. Finally, a summary of the chapter is presented in Section 6.6.

6.3 Experimental arrangement

Light from a 100 mW red laser diode is passed through a glass diffractive optical element (DOE) to generate a pattern of spots. Ten snap-shots of the pattern, which is projected on a portable projector screen placed approximately 2.2 m from the centre of the laser-DOE arrangement(LDA), are captured using a black and white fire-wire camera (POINTGREY RESEARCH FL2-08S2M) placed adjacent to the laser-DOE arrangement. The two co-ordinate systems, one fixed on the centre of the DOE and the other on the lens of the camera, can be associated with each other by

considering rectilinear shifts only (refer to the model developed in Chapter 5); such a 'stereo-rig' configuration [43] has been chosen to alleviate some of the difficulties of mapping the world co-ordinates of the laser spots incident on the screen with the co-ordinates of the spot's corresponding projections on the image plane of the camera. From the ten snap-shots the mean position of each of the spots on the camera image plane is estimated and the corresponding spot registered. A 21×21 sized box is drawn around each of the registered spot locations; the region encompassed by a box is searched periodically to determine the shift in the spot position which is then taken as an estimate of deformation. The deformation estimates corresponding to the shifted spots are returned to a separate routine that draws 17×17 pixel sized boxes around their initial mean locations; these boxes are filled with pixel intensities proportional to the corresponding spot deformation estimate.

Note that using the model equations developed in Chapter 5 and the estimated model parameters (Section 5.5 of Chapter 5), the resolution and the working range of the system with the screen kept at a distance of 2.2 m from the centre of the DOE were determined *a priori*. Resolution of the constructed system is defined as the change of depth of an object point that will result in a shift of its projection on the image plane by 1 pixel.

Using model equation (5.70) of Chapter 5 for the central spot we get:

$$-\rho + \left[\frac{-\varphi_x}{r - \varphi_z}\right](-f) = \left[\frac{-\varphi_x}{r_1 - \varphi_z}\right](-f) \text{ [Considering } (x_{PIobs2} - x_{PIobs1}) = -1] \tag{6.1}$$

Putting in equation (6.1) the model parameter values and substituting r by 2.2 m we get:

$$r_1 = 2.132\ m$$

So, resolution of the system at a range of 2.2 m is approximately $(2.2 - 2.132)\ m\ = 0.068\ m$.

The working range of a structured light based depth sensing system is determined by the maximum change of depth of an object point for which the position of its projection on the image plane can be located unambiguously. Since in the developed method a 21×21 box around the mean position of an object point's projection on

the image plane is searched to located the point's current projected position, the working range of the entire system is limited by a projection location shift of 10 pixels on the image plane.

Again using equation (5.70) of Chapter 5 for the central spot we get:

$$-10\rho + \left[\frac{-\varphi_x}{r-\varphi_z}\right](-f) = \left[\frac{-\varphi_x}{r_1-\varphi_z}\right](-f) \text{ [Considering } (x_{PIobs2} - x_{PIobs1}) = -10] \tag{6.2}$$

Substituting the model parameter values estimated in Section 5.5 of Chapter 5 and taking r as 2.2 m we get r_1 as:

$$r_1 = 1.67\ m$$

Thus the working range of the depth estimating set-up with the screen kept at a distance of 2.2 m away from the centre of the DOE is given by $(2.2-1.67)\ m = 0.53\ m$. It has been ensured while conducting the experiments that the subject stood inbetween the 1.67 m mark from the DOE plane and the screen to avoid ambiguity.

At first, the initial frame with no foreground object between the LDA and the screen is stored; subsequent frames are subtracted from it to generate a sequence of difference frames. A difference frame contains only those spots whose positions get shifted from their corresponding mean location on the image plane. Deformation estimates are generated for all the spots in the difference frame using their initial registered locations and their current positions within the corresponding 21×21 pixel sized boxes. As has been mentioned earlier, 17×17 pixel boxes are then drawn around the mean locations of the shifted spots, each filled with a flat intensity value generated using the corresponding spot's deformation estimate. However, it has been noted that the frames that come out of the process are usually contaminated with noisy pixel blocks. Noise creeps into the frames due to camera electronics and also due to other practical factors involved in generating deformation estimates. The output foreground object frames are finally de-noised by treating those with either of the two custom-made median filtering schemes that are developed in this chapter.

It should also be mentioned here that to generate binary masks of single or non-overlapping foreground objects true deformation estimates are not required. Instead,

identification of spots whose positions have shifted on the image plane is enough to generate the masks containing the target object(s).

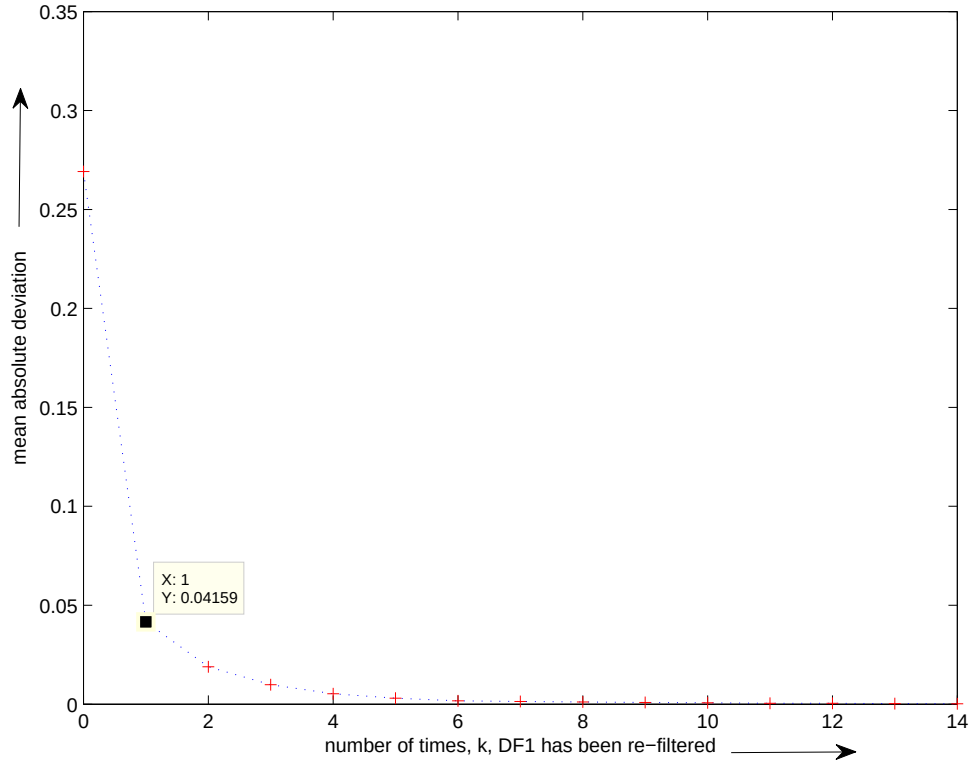


Figure 6.1: A typical $\hat{\delta}(k)$ vs k plot; note the plot has been obtained after filtering a difference frame, $DF1$, 15 times using the custom-made median filter.

6.4 Scheme 1

A custom-made median filter is designed to remove the isolated blocks in the difference frame and to fill up the discontinuities in the main object blob. The subsequence, $\Omega^{W_{ij}}$, spanned by the $(2N+1) \times (2N+1)$ $(N \in \mathbb{Z}^+, N \neq 0)$ window, W_{ij}, of the filter stationed on the $(i,j)^{th}$ pixel is given as:

$$\Omega^{W_{ij}} = \{J(i+\lambda_1, j+\lambda_2) \ : \ \lambda_1, \lambda_2 = \{-16, 0, 16\}\} \tag{6.3}$$

where, in equation (6.3), $\{J(\cdot,\cdot)\}$ is the 2-D image sequence and $0 \leq J(\cdot,\cdot) \leq 255$.

The filter-window has been constructed taking into account the fact that the centres of two adjacent blocks are 17 pixels apart. For a 2-D space, it is known that

many passes of the non-recursive form of the median filter, in general, yields a root signal that is invariant to median filtering [93], [94], [95]. Similar observations were made by re-filtering the sequence, several times, using the non-recursive form of the described custom-made median filter. A metric ($\hat{\delta}$) is defined to find the mean absolute deviation of the subsequent output:

$$\hat{\delta}(k) = \frac{1}{MN}\Sigma_{i=1}^{M}\Sigma_{i=1}^{N} \mid J_f^{k+1}(i,j) - J_f^{k}(i,j) \mid \tag{6.4}$$

where, in equation (6.4), $J_f^k(i,j)$ denotes the output after re-filtering the sequence k times, and $M \times N$ is the size of the frame.

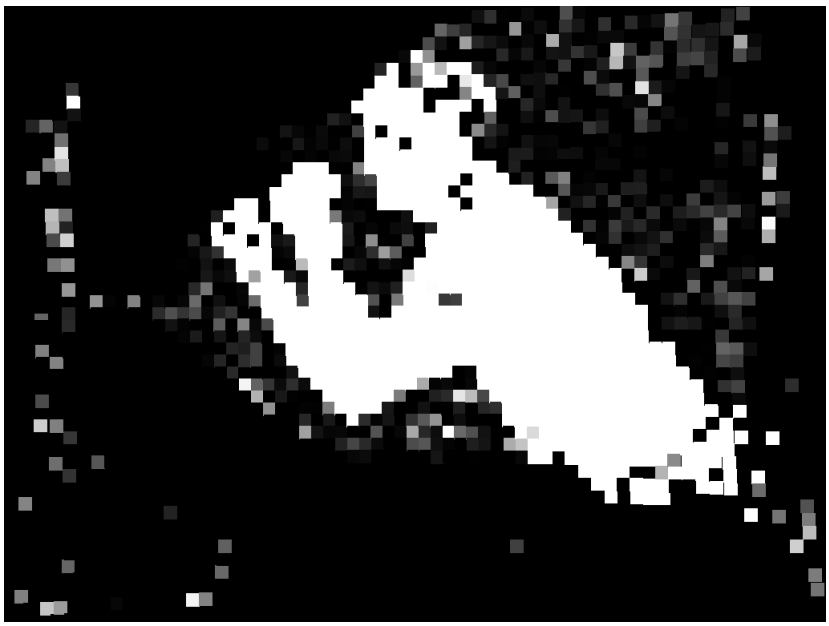
Fig. 6.1 shows a typical variation of $\hat{\delta}(k)$ with increments in k; it is evident that the mean of the absolute deviation between the output frames gets smaller and smaller with increments in k, an indication that a step closer to the root signal is reached with every pass of the filter.
A modified median filter has also been designed to generate the final result of the process; this filter is triggered after filtering the sequence k times when $(\hat{\delta}(k)/\hat{\delta}(1))$ is less than some pre-specified threshold, $\tau_{\hat{\delta}}$; note, the speed of the entire process depends on the choice of the threshold.

The modified median filter estimates the sum of the intensities of the pixels encompassed by a window, W_{ij}. If the sum of the pixel intensities is above a chosen threshold, then the $(i,j)^{th}$ pixel intensity is replaced by the output of the filter described above, otherwise it is forced to be 0.

6.4.1 Results obtained using Scheme 1

A person stood in-between the laser-camera arrangement and the screen taking different poses. The upper part of his body was segmented using the developed custom-made median filtering scheme. The output difference frames generated were contaminated with noisy pixel blocks. Two such difference frames, $DF1$ and $DF2$, are shown in Fig. 6.2(a) and Fig. 6.2(b). The results obtained after applying the custom-made median filtering scheme and repeating the process 14 times are shown in Fig. 6.3 — Fig. 6.18 (excluding figures 6.10(ii), 6.11(i) and 6.18(ii)). Note that all

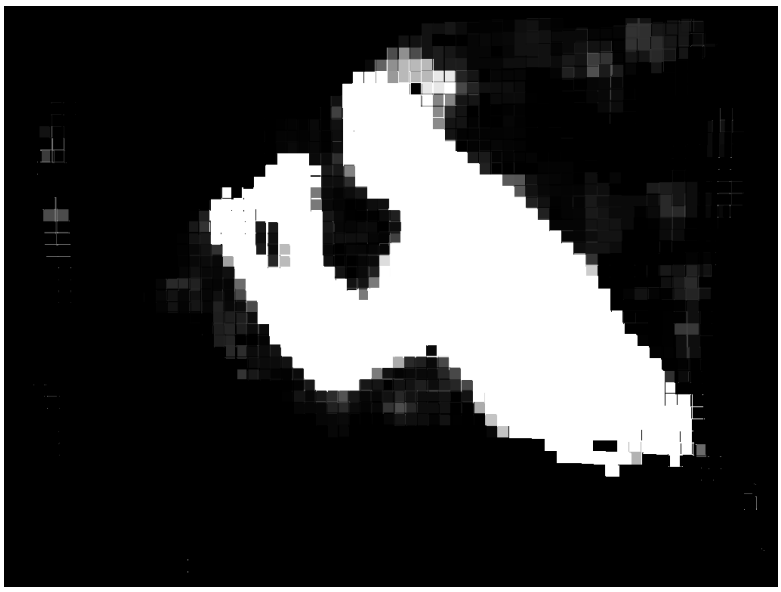
the figures after Fig. 6.3 till Fig. 6.18(i) (inclusive) have been labelled in their respective figure captions using the difference frame identifier ($DF1$ or $DF2$), the number of times it has re-filtered, k, the value of the defined metric, $\hat{\delta}(k)$, and the ratio $\frac{\hat{\delta}(k)}{\hat{\delta}(1)}$. The final results obtained after applying the modified median filter are shown in Fig. 6.10(ii) and Fig. 6.18(ii). It is noted that the designed median-filtering scheme meets its objectives; however, it should be kept in mind that applying a median filtering scheme with a large kernel size is a time consuming process and so to optimise speed the value of the threshold, $\tau_{\hat{\delta}}$, should be chosen with deliberation. It becomes apparent from Fig. 6.1 and Fig. 6.3 — Fig. 6.18(i) that the re-filtering process can be stopped after $k = 5$ $\left[\frac{\hat{\delta}(5)}{\hat{\delta}(1)} = 0.0112\ (for\ DF1);\ \frac{\hat{\delta}(5)}{\hat{\delta}(1)} = 0.0080\ (for\ DF2)\right]$ and the modified filter applied to generate the final result.



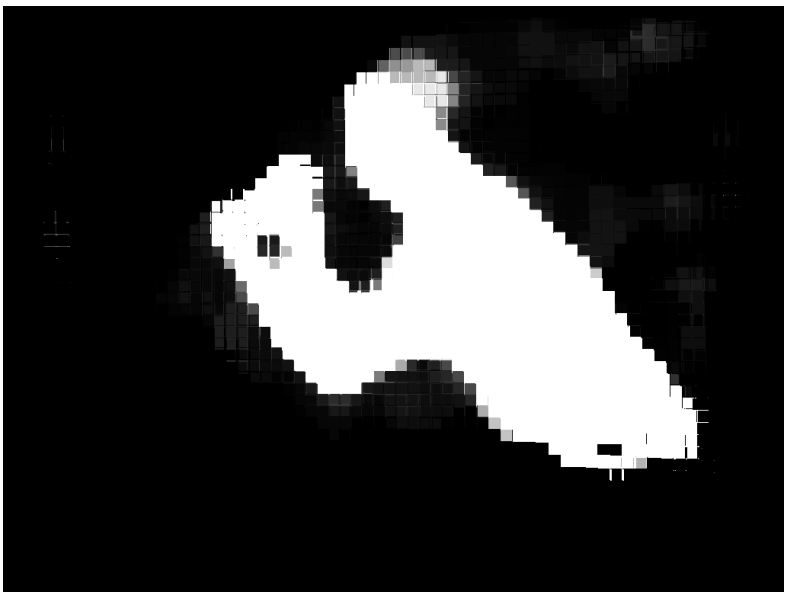
(a) — $DF1$

(b) — $DF2$

Figure 6.2: Two typical difference frames generated using the deformation based foreground object segmentation method; difference frame (a) is labelled as $DF1$ and difference frame (b) as $DF2$. Note both the difference frames are contaminated with noisy pixel blocks.

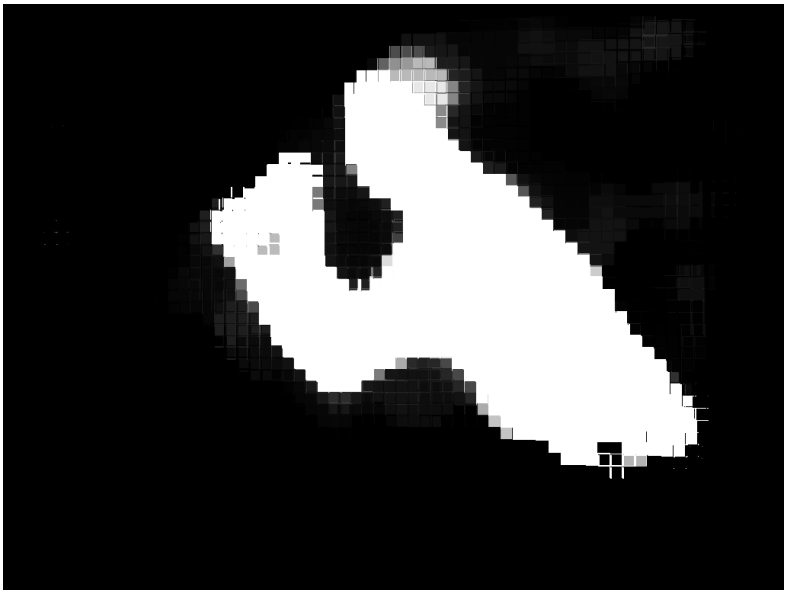


(i)

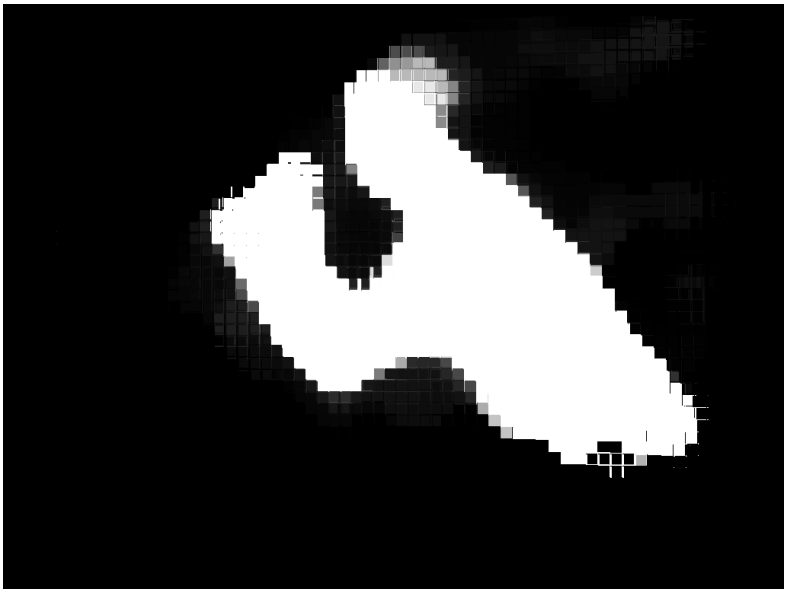


(ii)

Figure 6.3: (i): the result after applying the custom-made median filtering scheme on $DF1$; (ii) the result after re-filtering the image with the same custom-made filter $\left[\hat{\delta}(1) = 0.0416, \frac{\hat{\delta}(1)}{\hat{\delta}(1)} = 1\right]$.

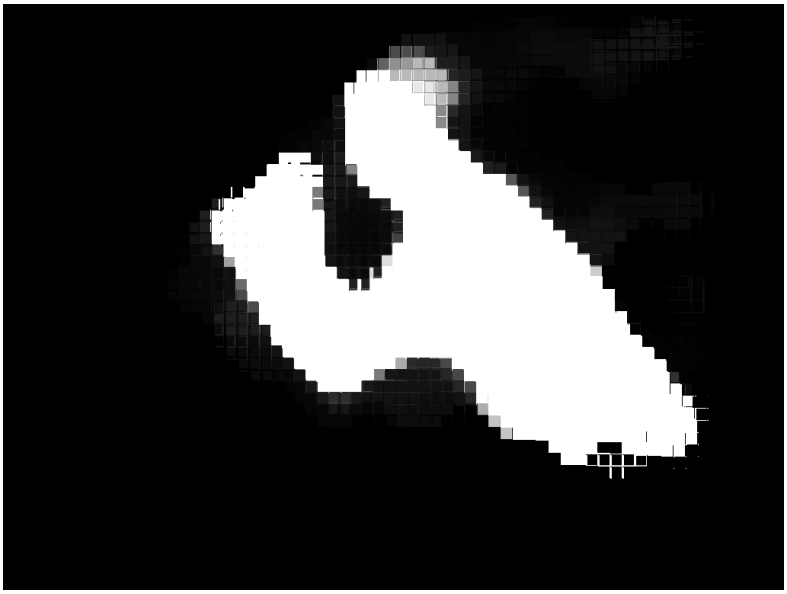


(i)

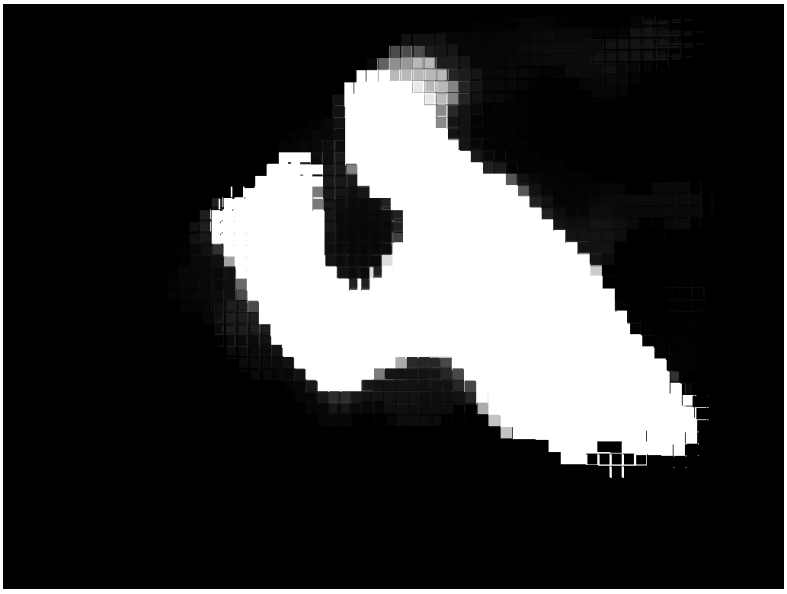


(ii)

Figure 6.4: (i) $DF1$, $k = 2$, $\hat{\delta}(2) = 0.0189$, $\frac{\hat{\delta}(2)}{\hat{\delta}(1)} = 0.0701$; (ii) $DF1$, $k = 3$, $\hat{\delta}(3) = 0.0099$, $\frac{\hat{\delta}(3)}{\hat{\delta}(1)} = 0.0367$.



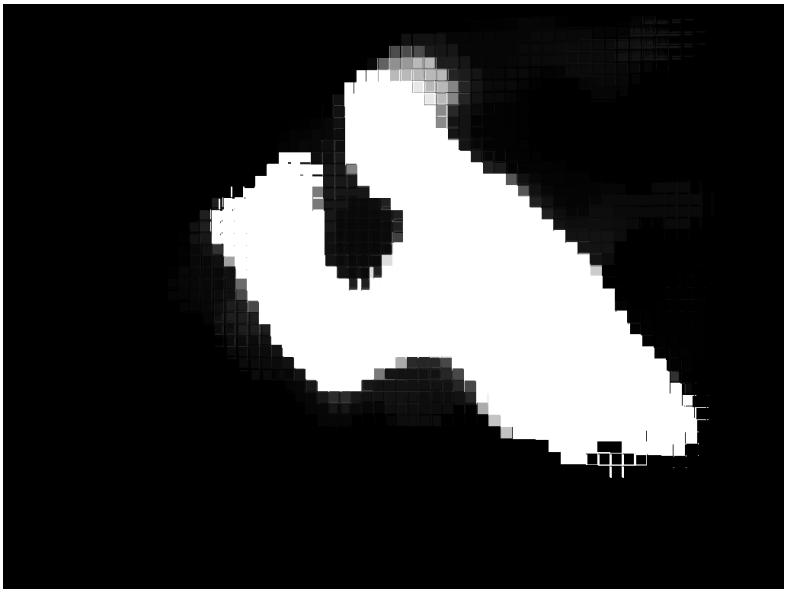
(i)



(ii)

Figure 6.5: (i) $DF1$, $k = 4$, $\hat{\delta}(4) = 0.0053$, $\frac{\hat{\delta}(4)}{\hat{\delta}(1)} = 0.0196$; (ii) $DF1$, $k = 5$, $\hat{\delta}(5) = 0.0030$, $\frac{\hat{\delta}(5)}{\hat{\delta}(1)} = 0.0112$.

(i)



(ii)

Figure 6.6: (i) $DF1$, $k = 6$, $\hat{\delta}(6) = 0.0017$, $\frac{\hat{\delta}(6)}{\hat{\delta}(1)} = 0.0062$; (ii) $DF1$, $k = 7$, $\hat{\delta}(7) = 0.0014$, $\frac{\hat{\delta}(7)}{\hat{\delta}(1)} = 0.0051$.

(i)

(ii)

Figure 6.7: (i) $DF1$, $k = 8$, $\hat{\delta}(8) = 0.0033$, $\frac{\hat{\delta}(8)}{\hat{\delta}(1)} = 0.0042$; (ii) $DF1$, $k = 9$, $\hat{\delta}(9) = 8.7978 \times 10^{-4}$, $\frac{\hat{\delta}(9)}{\hat{\delta}(1)} = 0.0033$.

(i)

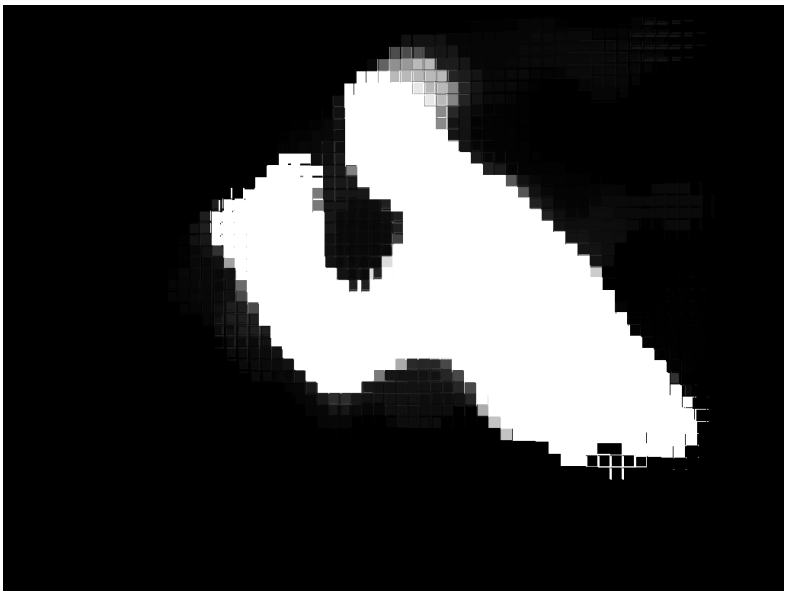
(ii)

Figure 6.8: (i) $DF1$, $k = 10$, $\hat{\delta}(10) = 6.7186 \times 10^{-4}$, $\frac{\hat{\delta}(10)}{\hat{\delta}(1)} = 0.0025$; (ii) $DF1$, $k = 11$, $\hat{\delta}(11) = 4.9769 \times 10^{-4}$, $\frac{\hat{\delta}(11)}{\hat{\delta}(1)} = 0.0018$.

(i)

(ii)

Figure 6.9: (i) $DF1$, $k = 12$, $\hat{\delta}(12) = 4.2528 \times 10^{-4}$, $\frac{\hat{\delta}(12)}{\hat{\delta}(1)} = 0.0016$; (ii) $DF1$, $k = 13$, $\hat{\delta}(13) = 3.3041 \times 10^{-4}$, $\frac{\hat{\delta}(13)}{\hat{\delta}(1)} = 0.0012$.

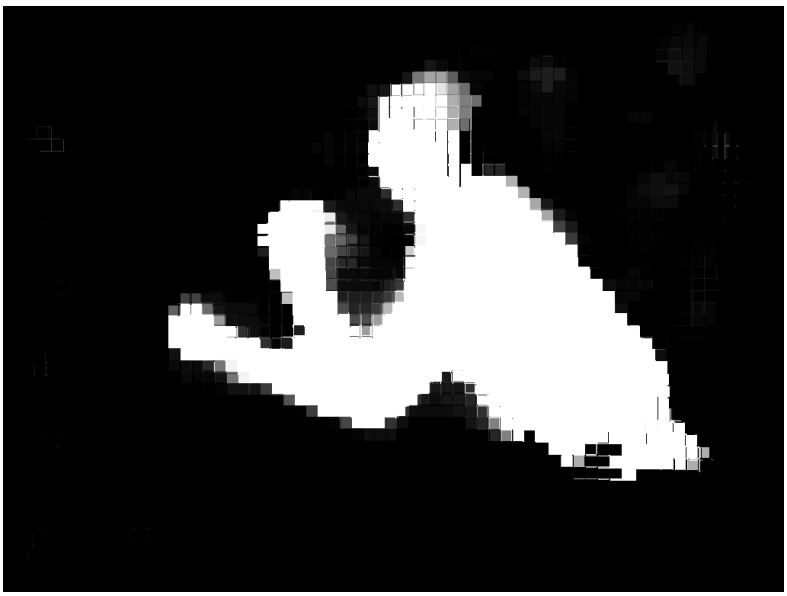


(i)

(ii)

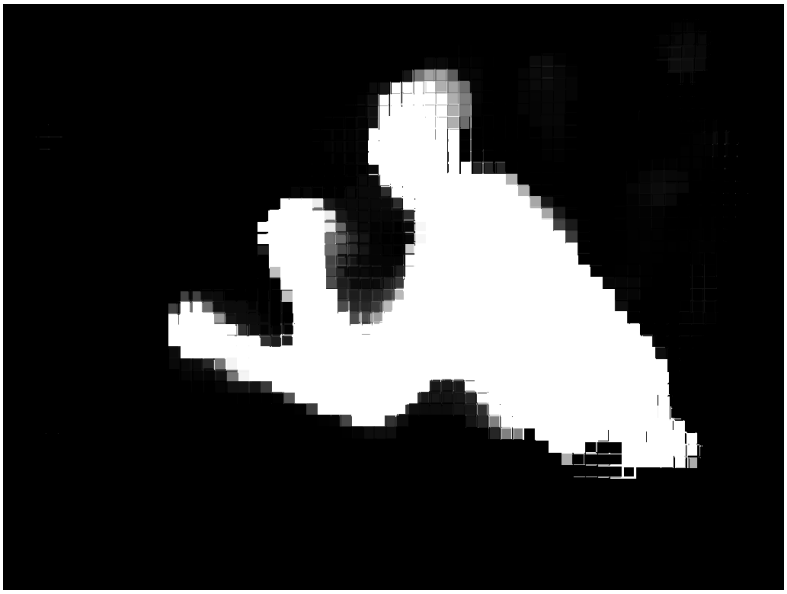
Figure 6.10: (i) $DF1$, $k = 14$, $\hat{\delta}(14) = 2.1350 \times 10^{-4}$, $\frac{\hat{\delta}(14)}{\hat{\delta}(1)} = 0.0008$; (ii) The result after applying the modified median filter on $DF1$ after filtering it 15 times using the custom-made median filter.

(i)

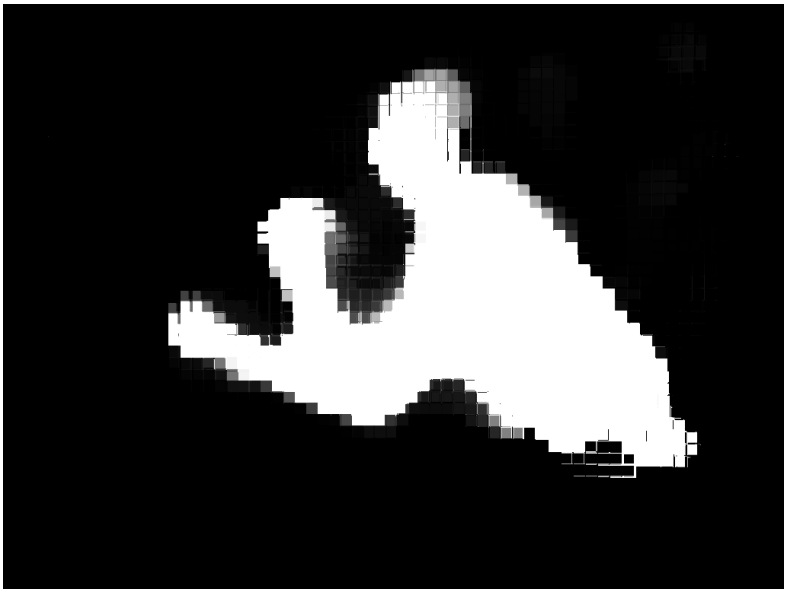


(ii)

Figure 6.11: (i) The result after applying the custom-made median filter on $DF2$. The next few images show the effects of re-filtering the previous output of the filter with the same filter. Again, as before the figures will be labelled in the figure captions using the difference frame identifier, the number of times it has been re-filtered, k, the value of the metric $\hat{\delta}(k)$ and the ratio $\frac{\hat{\delta}(k)}{\hat{\delta}(1)}$; (ii) $DF2$, $k = 1$, $\hat{\delta}(1) = 0.0565$, $\frac{\hat{\delta}(1)}{\hat{\delta}(1)} = 1.000$.

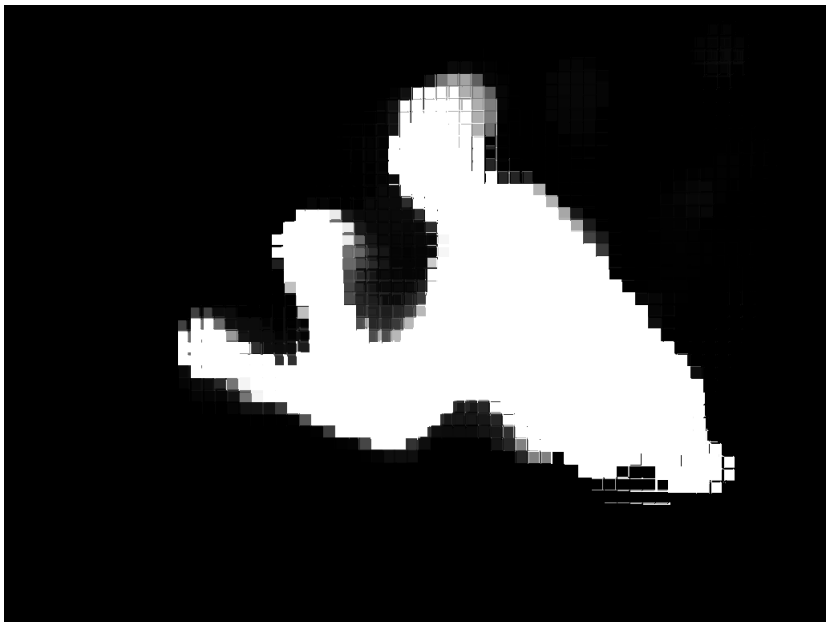


(i)

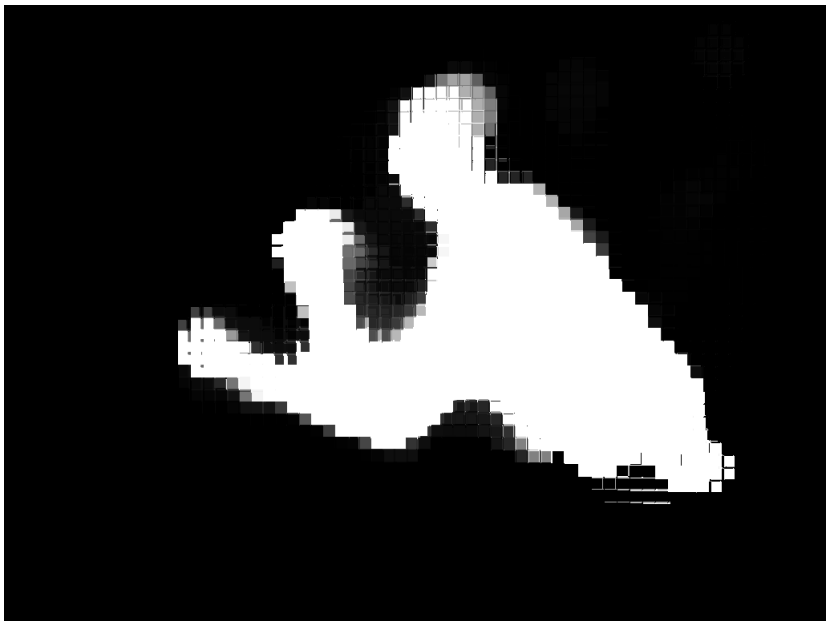


(ii)

Figure 6.12: (i) $DF2$, $k = 2$, $\hat{\delta}(2) = 0.0198$, $\frac{\hat{\delta}(2)}{\hat{\delta}(1)} = 0.0511$; (ii) $DF2$, $k = 3$, $\hat{\delta}(3) = 0.0033$, $\frac{\hat{\delta}(3)}{\hat{\delta}(1)} = 0.0253$.

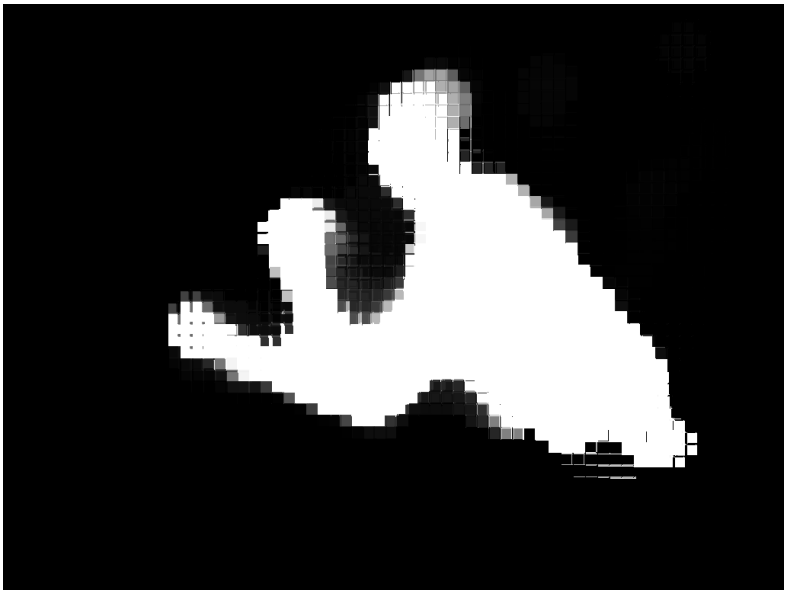


(i)

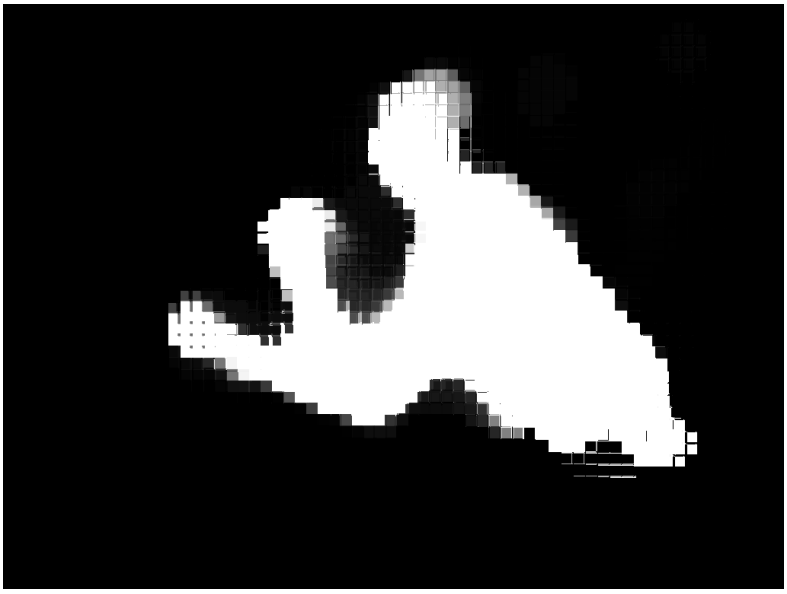


(ii)

Figure 6.13: (i) $DF2$, $k = 4$, $\hat{\delta}(4) = 0.0051$, $\frac{\hat{\delta}(4)}{\hat{\delta}(1)} = 0.0132$; (ii) $DF2$, $k = 5$, $\hat{\delta}(5) = 0.0031$, $\frac{\hat{\delta}(5)}{\hat{\delta}(1)} = 0.0080$.

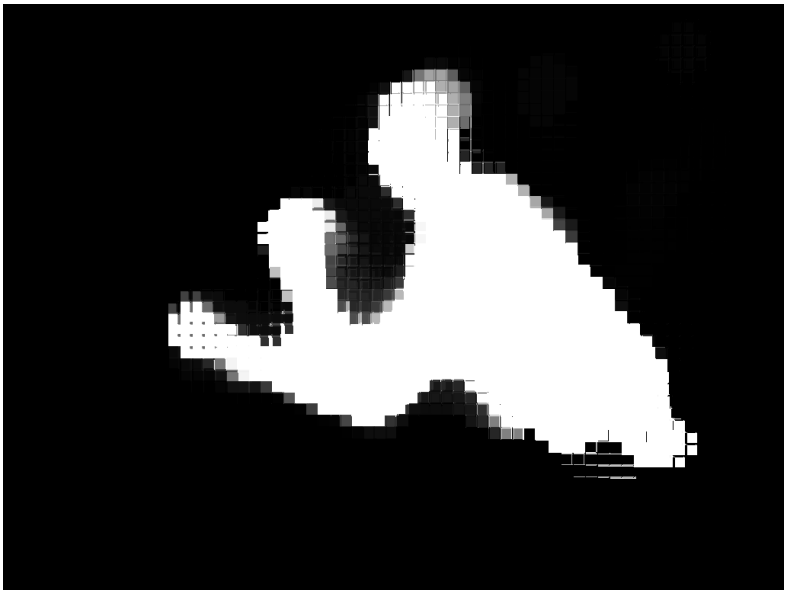


(i)

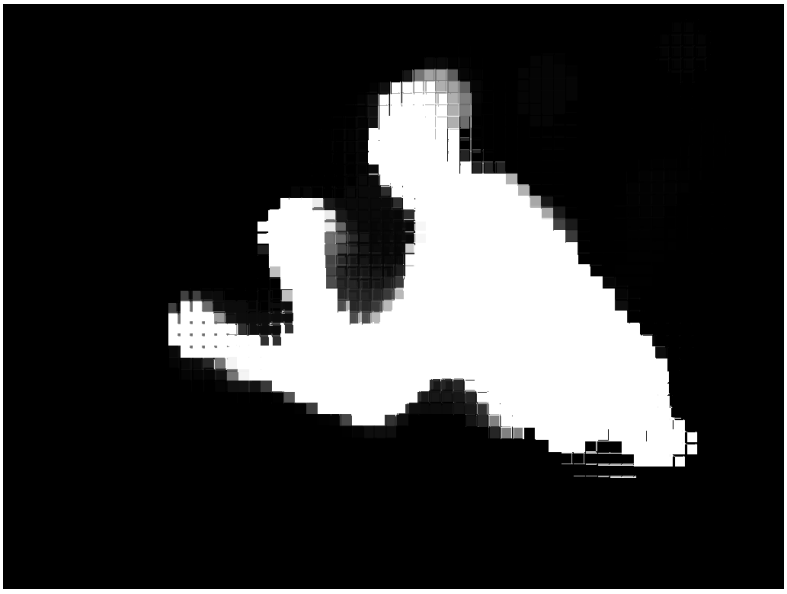


(ii)

Figure 6.14: (i) $DF2$, $k = 6$, $\hat{\delta}(6) = 0.0021$, $\frac{\hat{\delta}(6)}{\hat{\delta}(1)} = 0.0053$; (ii) $DF2$, $k = 7$, $\hat{\delta}(7) = 0.0018$, $\frac{\hat{\delta}(7)}{\hat{\delta}(1)} = 0.0046$.

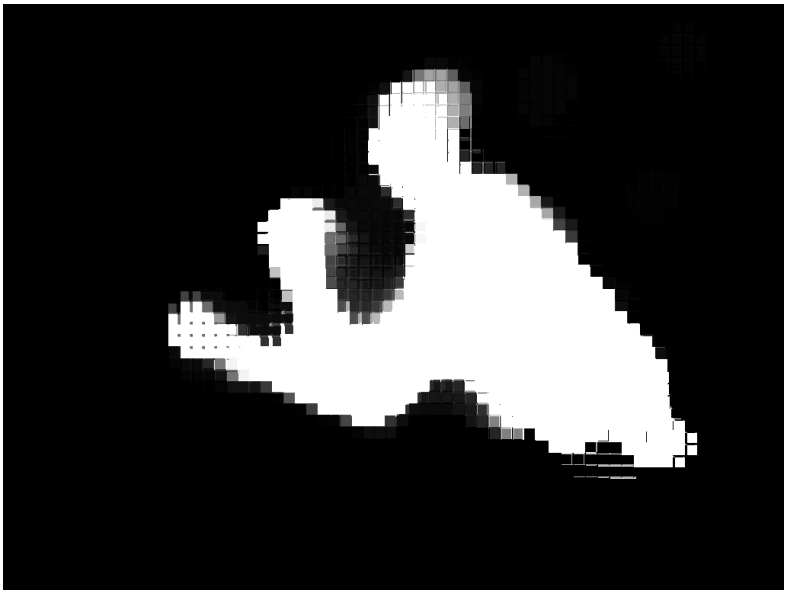


(i)

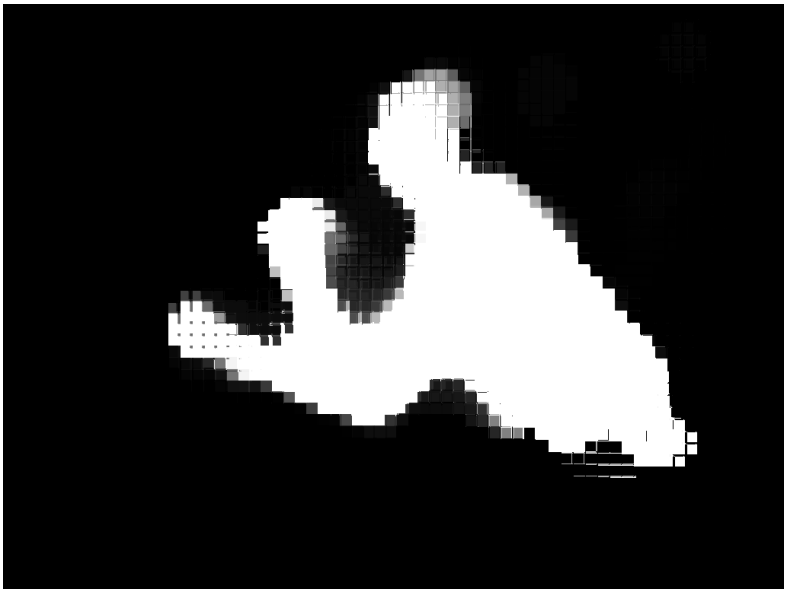


(ii)

Figure 6.15: (i) $DF2$, $k = 8$, $\hat{\delta}(8) = 0.0014$, $\frac{\hat{\delta}(8)}{\hat{\delta}(1)} = 0.0035$; (ii) $DF2$, $k = 9$, $\hat{\delta}(9) = 0.0011$, $\frac{\hat{\delta}(9)}{\hat{\delta}(1)} = 0.0028$.

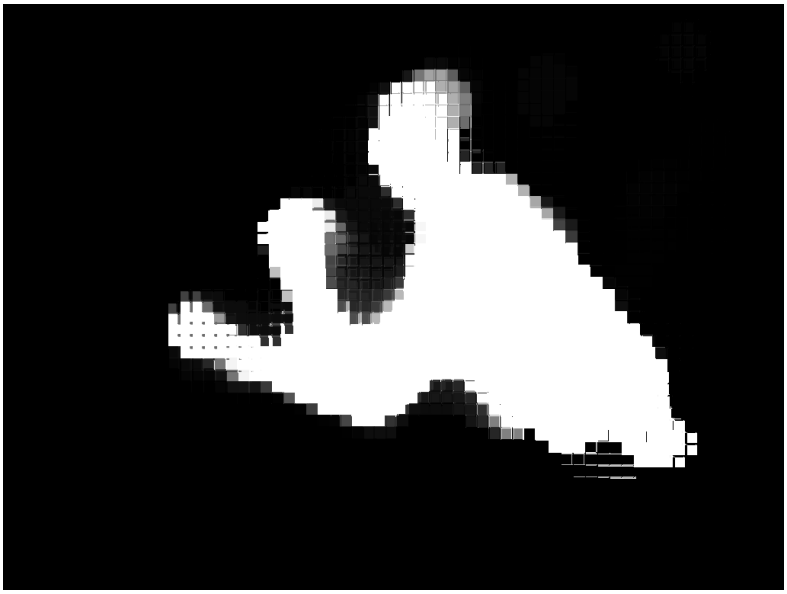


(i)

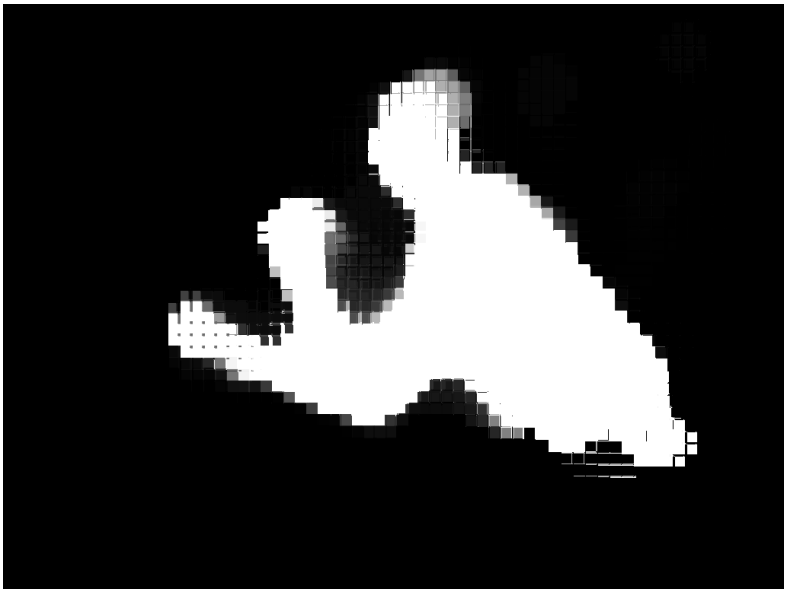


(ii)

Figure 6.16: (i) $DF2$, $k = 10$, $\hat{\delta}(10) = 8.5352 \times 10^{-4}$, $\frac{\hat{\delta}(10)}{\hat{\delta}(1)} = 0.0022$; (ii) $DF2$, $k = 11$, $\hat{\delta}(11) = 6.5161 \times 10^{-4}$, $\frac{\hat{\delta}(11)}{\hat{\delta}(1)} = 0.0017$.



(i)



(ii)

Figure 6.17: (i) $DF2$, $k = 12$, $\hat{\delta}(12) = 5.3490 \times 10^{-4}$, $\frac{\hat{\delta}(12)}{\hat{\delta}(1)} = 0.0014$; (ii) $DF2$, $k = 13$, $\hat{\delta}(13) = 3.8640 \times 10^{-4}$, $\frac{\hat{\delta}(13)}{\hat{\delta}(1)} = 0.0010$.

6.5 Scheme 2

This scheme makes use of the multistage max/median and min/median filters [96], [97], [98] to cover up the discontinuities in the target blob, and to remove the noisy blocks from the difference frame. Two unidirectional filter windows, $W^1_{i,j}$ and W^2_{ij} with the window axes orthogonal to each other are chosen as the subfilter windows. Sub-sequences, $\Omega^{W^l_{ij}}$, $l \in \{1,2\}$, spanned by the two windows are defined as follows:

$$\begin{aligned}\Omega^{W^1_{ij}} &= \{J(i+\lambda_1, j+\lambda_2) : -N \leq \lambda_1 \leq N, \lambda_2 = 0\} \\ \Omega^{W^2_{ij}} &= \{J(i+\lambda_1, j+\lambda_2) : -N \leq \lambda_2 \leq N, \lambda_1 = 0\}\end{aligned} \tag{6.5}$$

The output of the sub-filters, Y^l_{ij}, $l \in \{1,2\}$, are defined as:

$$Y^l_{ij} = median\{J(\cdot,\cdot) : J(\cdot,\cdot) \in \Omega^{W^l_{ij}}\},\ l \in \{1,2\} \tag{6.6}$$

The output of the max/median and min/median filters as defined in [98] are as follows:

$$\begin{aligned}Y^{max/median}_{ij} &= max\{Y^l_{ij}\},\ l \in \{1,2\} \\ Y^{min/median}_{ij} &= min\{Y^l_{ij}\},\ l \in \{1,2\}\end{aligned} \tag{6.7}$$

6.5.1 Choice of the variable N

Since the multistage max/min filter designed to smear an isolated 17×17 block in the difference frame is made up of two orthogonal unidirectional sub-filters, well understood $1-$D median filter properties have been used to determine the value of N. It follows from the basic properties of a $1-$D median filter that strictly monotonic $1-$D sequences come out invariant to median filtering implemented with a window of any arbitrary length. However, since a $1-$D median filter is always implemented with a fixed and finite length window, the requirement of strict monotonicity is unnecessarily restrictive; a signal will come out invariant to median filtering if every segment of it, as the filter window is passed through it, is monotonic in nature. However, even this requirement can be further relaxed as elaborated below [†] :

A sequence $\{\vartheta(n)\}$ is defined as locally monotonic of length ℓ, $[LOMO(\ell)]$,

[†] the theory is based on the studies on median filter properties as reported in [93], [94] and [95]

if $(\vartheta(n), \ldots, \vartheta(n+\ell-1))$ is monotonic for each n.

§ **Observation**#01**:** A $LOMO(\ell)$ sequence is also $LOMO(\iota)$ if $\iota < \ell$.

§ **Observation**#02**:** Let $\{\vartheta(n)\}$ be a $LOMO(\ell)$ sequence. Then both segments: $(\vartheta(n), \ldots, \vartheta(n+\ell-1))$ and $(\vartheta(n+1), \ldots, \vartheta(n+\ell))$ are monotonic. Now if $\vartheta(n) < \vartheta(n+\ell-1)$ and $\vartheta(n+1) > \vartheta(n+\ell)$, then $\vartheta(i) \leqslant \vartheta(j)$ and $\vartheta(i) \geqslant \vartheta(j)$ $\forall$ $(n+1) \leqslant i, j \leqslant (n+\ell-1)$; this means $\vartheta(n+1) = \vartheta(n+2) \ldots = \vartheta(n+\ell-1)$.

Lemma 1: Therefore, an alternative definition of a $LOMO(\ell)$ sequence that relaxes the root-sequence requirements is that if there is any change in trend, a $LOMO(\ell)$ sequence must stay constant for $(\ell-1)$ samples.

Theorem 1: In general, a $LOMO(\ell)$ sequence is a root sequence of a median filter implemented with a window of length $(2N+1)$ if $N \leqslant (\ell-1)$.

Proof: Consider a segment of length $(2N+1)$: $(x(n-N), \ldots, x(n+N))$. This segment is either monotonic or, if there is a change in trend, the sequence remains constant for $(N+1) = (\ell-1)$ samples (according to the alternative definition of a $LOMO(\ell)$ sequence). In the first case it is apparent that $x(n)$ is the median of the segment. In the second case since the length of the segment that stays constant is $(N+1)$, there are at least $(N+1)$ elements in the segment equal to $x(n)$. So, the median in the second case is also $x(n)$.

Note that the relaxed definition of a $LOMO(\ell)$ sequence (Lemma 1) and Theorem 1 have been used to determine the size of the subfilters in this subsection.

Recall one of the main purposes of applying the multistage max/median and min/median filters is to smear isolated blocks of size 17×17. This, in turn, means that the purpose of a unidirectional sub-filter is to smudge isolated $1-$D segments that stay constant for 17 pixels. At first, the maximum length, $(2N+1)_{max}$, of a sub-filter window is determined that will do just the opposite i.e. preserve the isolated pat-

terns. The fact that choosing a sub-filter window that is larger than $(2N+1)_{max}$ will smear the pattern is intuitively apparent.

From the alternative definition of a $LOMO(\ell)$ sequence it is known that if there is any change of pattern, a $LOMO(\ell)$ sequence has to stay constant for $(\ell-1)$ samples. It is assumed that each row (or column) of a difference frame is a $LOMO(18)$ sequence ($\ell = 18$) in which even if there is any change of pattern, $(\ell-1) = 17$ samples remain constant. Now from Theorem 1 it is known that a 1−D median filter implemented with a window of length $(2N+1)$ will preserve a $LOMO(\ell)$ sequence if $N \leqslant (\ell-2)$ i.e. $N \leqslant 16$ in this case ($\because \ell = 18$). So, the maximum length of a sub-filter window that will preserve the pattern is $(2\times 16+1) = 33$ i.e. $(2N+1)_{max} = 33$. So, to smear a 17 pixel long isolated segment, the sub-filters must be implemented with a window length greater than 33 i.e. $(2N+1) > 33$. In the experiments conducted the orthogonal sub-filters were implemented with a window size of 37. For the multistage max/median filter this means that the filter will preserve patterns that exit in either of the two orthogonal directions for atleast two 17×17 blocks. Finally, a word of caution. While selecting the size of a sub-filter window it should be borne in mind that choosing a window of large dimension not only might smear some sections of the real target blob but also will reduce the speed of the entire process.

6.5.2 Procedure

At first the difference frame is filtered using a multistage max/median filter using the two defined sub-filters with window length of 37 each. Treating the difference image with the max/median filter helps in bridging the discontinuities in the foreground object blob, and helps to preserve any feature that exists for more than 17 pixels in any of the two orthogonal directions. Note, this preserves the positive characteristics of the image as has been mentioned in [98]. A binary-mask is then generated from the intensity image by choosing a low threshold. A multistage min/median filter is then applied to the binary mask to remove most of the remaining isolated blocks. Finally, holes in the main binary blob are filled, and an area-filter is deployed to remove the remaining isolated blobs which are not a part of the target object.

6.5.3 Results obtained using Scheme 2

The two difference frames, $DF1$ and $DF2$, shown in Fig. 6.2(a) and Fig. 6.2(b), were treated with the sequence of methods outlined in Section 6.5.2 of Scheme 2. The output frames are shown in Fig. 6.19 and 6.20. The results obtained are better than those obtained through the application of Scheme 1. In addition to this, it should also be borne in mind that Scheme 2 has another edge over Scheme 1. To implement Scheme 2 no difficult thresholds need to be chosen. However, to deploy Scheme 1 thresholds to stop the re-filtering process and control the working of the modified median filter have to be selected with deliberation after extensive testing.

6.5.4 Analogy with morphological operators

Note that the application of the multistage max/median filter and the min/median filter can be thought of as treating the difference image with some basic morphological operators. Recall the two basic mathematical morphology transformations are erosion and dilation of J by W, where $\{J(\cdot,\cdot)\}$ is the 2-D sequence, and W the structuring element [97], [99], [100], [101]. The output of the erosion operator $[J \ominus W]$, as given in [97], at the $(i,j)^{th}$ position and can be expressed as:

$$[J \ominus W](i,j) = \min_{(\lambda_1,\lambda_2)\in W}\{J(i+\lambda_1, j+\lambda_2)\} \tag{6.8}$$

It means the output of the operator is the minimum sample value of the samples spanned by the window W. In the same way the output of a discrete dilation operation operation at the $(i,j)^{th}$ position is given by:

$$[J \oplus W](i,j) = \max_{(\lambda_1,\lambda_2)\in W}\{J(i+\lambda_1, j+\lambda_2)\} \tag{6.9}$$

The working of the multistage max/median filter and the min/median filter in the light of the basic morphological operations can be described as:

$$Y^l_{i,j} = median(\Big\{J(\cdot,\cdot) : J(\cdot,\cdot) \in \Omega^{W^l_{ij}}\Big\}) \tag{6.10}$$

$$= \max(\Big\{\left[J \ominus W^l_{N+1}\right]^{\Diamond N+1}(i,j)\Big\}) \tag{6.11}$$

where, in equation (6.11), l represents the direction of the structuring element (sub-filter window), $\Diamond N+1$ the fact that the operation is repeated $(N+1)$ times and W^l_{N+1} represents a window in the direction l that spans $(N+1)$ elements. It is also

to be ensured that every time the operation is repeated, one of the sample points spanned by the window during its previous occurrence has to be excluded; moreover, a sample point once excluded cannot be included again while selecting $(N+1)$ sample points out of the $(2N+1)$ sample points for another run of the operation. This can be accomplished using a sliding window of length $(N+1)$ spanning the first $(N+1)$ sample points of the structural element when the operation is initiated and then shifting it by one place for every repetition. It follows immediately that when the operation is repeated $(N+1)$ times, N sample points will be excluded and all $(2N+1)$ sample points in a specific direction (encompassed by a sub-filter window) will be scanned. Note that the erosion operations can also be replaced by dilations and then the output of a sub-filter can be expressed as:

$$Y_{ij}^{l} = min(\Big\{ \left[J \oplus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\}) \tag{6.12}$$

The final output of the multistage max/median filter can be expressed as:

$$\begin{aligned} Y_{ij}^{max/median} &= \max_{l}(\{Y_{ij}^{l}\}) && (6.13)\\ &= \max_{l}(\max(\Big\{ \left[J \ominus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\})) && (6.14)\\ &= \max(\bigcup_{l}(\Big\{ \left[J \ominus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\})) && (6.15)\\ &= \max_{l}(\min(\Big\{ \left[J \oplus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\})) \text{ [follows from (6.12)]} && (6.16) \end{aligned}$$

Note from one of the intermediate expressions [equation (6.15)] that only a *max* finding operation on a set formed by the union of the sets of erosions in different directions gives the final output of the multistage max/median filter. In a similar way the following equations outline the working of a multistage min/median filter in terms of the basic discrete morphological operations:

$$\begin{aligned} Y_{ij}^{min/median} &= \min_{l}(\{Y_{ij}^{l}\}) && (6.17)\\ &= \min_{l}(\max(\Big\{ \left[J \ominus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\})) && (6.18)\\ &= \min_{l}(\min(\Big\{ \left[J \oplus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\})) && (6.19)\\ &= \min(\bigcup_{l}(\Big\{ \left[J \oplus W_{N+1}^{l} \right]^{\Diamond N+1} (i,j) \Big\})) && (6.20) \end{aligned}$$

From this section it becomes apparent that the output of a multistage max/median filter or a multistage min/median can be determined through a series of *max* and/or *min* finding operations. Expressing the filter output in this way also makes the full complex sorting operation obsolete. Thus these equations can be used to design some dedicated architecture that can speed up the overall process of treating the difference frame with the described median filtering scheme.

Concluding the chapter, equations specific to the Scheme 2 median filtering schemes are given below:

$$\begin{aligned} Y_{ij}^{max/median} &= \max_{l}(\{Y_{ij}^{l}\}), && l \in \{1,2\} \\ &= \max_{l}(\max(\left\{ \left[J \ominus W_{19}^{l}\right]^{\Diamond 19}(i,j) \right\})), && l \in \{1,2\} \\ &= \max(\bigcup_{l}(\left\{ \left[J \ominus W_{19}^{l}\right]^{\Diamond 19}(i,j) \right\})), && l \in \{1,2\} \\ &= \max_{l}(\min(\left\{ \left[J \oplus W_{19}^{l}\right]^{\Diamond 19}(i,j) \right\})), && l \in \{1,2\} \end{aligned} \tag{6.21}$$

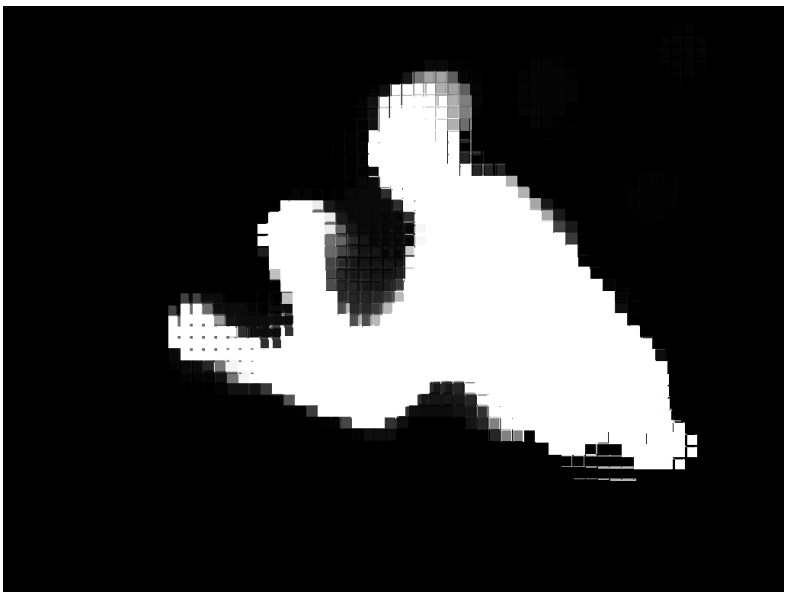
$$\begin{aligned} Y_{ij}^{min/median} &= \min_{l}(\{Y_{ij}^{l}\}), && l \in \{1,2\} \\ &= \min_{l}(\max(\left\{ \left[J \ominus W_{19}^{l}\right]^{\Diamond 19}(i,j) \right\})), && l \in \{1,2\} \\ &= \min_{l}(\min(\left\{ \left[J \oplus W_{19}^{l}\right]^{\Diamond 19}(i,j) \right\})), && l \in \{1,2\} \\ &= \min(\bigcup_{l}(\left\{ \left[J \oplus W_{19}^{l}\right]^{\Diamond 19}(i,j) \right\})), && l \in \{1,2\} \end{aligned} \tag{6.22}$$

6.6 Summary

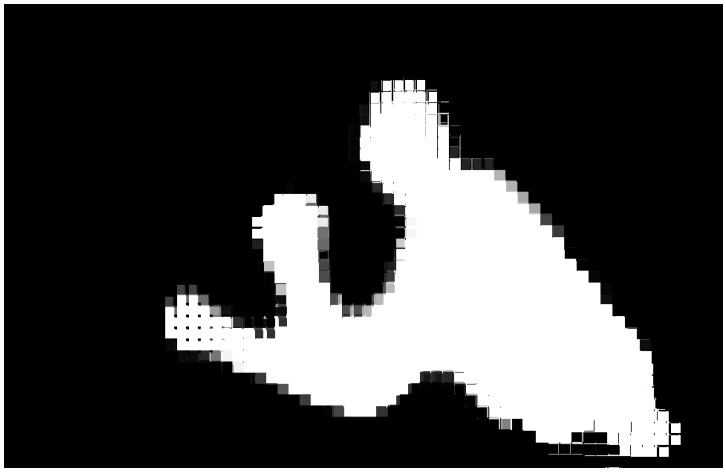
A description of an arrangement has been given in this chapter that can be used to segment foreground objects in a scene through the determination of imaged spot position disparities on the sensor plane of a camera. Rough foreground object silhouettes can be generated using the described set up as it draws 17×17 pixel blocks around the mean position of all the spots that shift from their corresponding mean initial positions on the image plane. Each of the blocks are then filled with flat intensity values proportional to the corresponding spot deformation estimate. It should be noted the deformation estimate is generated after calculating the Euclidean distance between a shifted spot's initial and final locations within a 21×21 pixel sized box drawn around the spot's mean initial location. Though explicit depth

maps are not generated by the developed process, deformation estimates obtained through the use of it can still be used to segment multiple partially overlapping or non-overlapping objects or a single object situated between the LDA and the screen. The focus of this chapter has, however, been further limited to the generation of non-overlapping or single object(s) masks.

It has been noted that the difference frames (masks) generated through the use of the described set-up are usually contaminated with noisy pixel clusters. Two median filtering schemes have also been developed in this chapter to de-noise the difference frames. The first scheme keeps re-filtering a difference frame using a custom-made median filter until the value of a defined metric goes below a pre-specified threshold; it then triggers a modified median filter to generate the final result. The results obtained through the use of the scheme look satisfactory. It not only removes the noisy pixel clusters from the difference frames but also bridges the discontinuities in the real target blob(s). However, the downsides of using the scheme include selection of two difficult thresholds, one to stop the time consuming re-filtering process and the other to control the working of the modified median filter.
The other scheme developed in this chapter makes use of multistage max/median and min/median filters to de-noise the noisy blocks contaminated difference frames. Both the multistage filters are implemented with two unidirectional orthogonal windows. The length of the windows have been determined through the use of well understood $1-D$ median filter properties. The subjective quality of the results obtained through the use of the second scheme is better than those obtained through the use of the first one. Moreover, no difficult thresholds need to be chosen to implement the second scheme. Finally, an analogy of the multistage max/median and min/median filters have been included in this chapter that might help speed up the second scheme through the design and development of a dedicated architecture.



(i)



(ii)

Figure 6.18: (i) $DF2$, $k = 14$, $\hat{\delta}(14) = 2.8015 \times 10^{-4}$, $\frac{\hat{\delta}(14)}{\hat{\delta}(1)} = 0.0007$; (ii) The result after applying the modified median filter on $DF2$ that has been filtered 15 times using the custom-made median filtering scheme.

Figure 6.19: The result after treating $DF1$ with the median filtering scheme elaborated in Section 6.5.

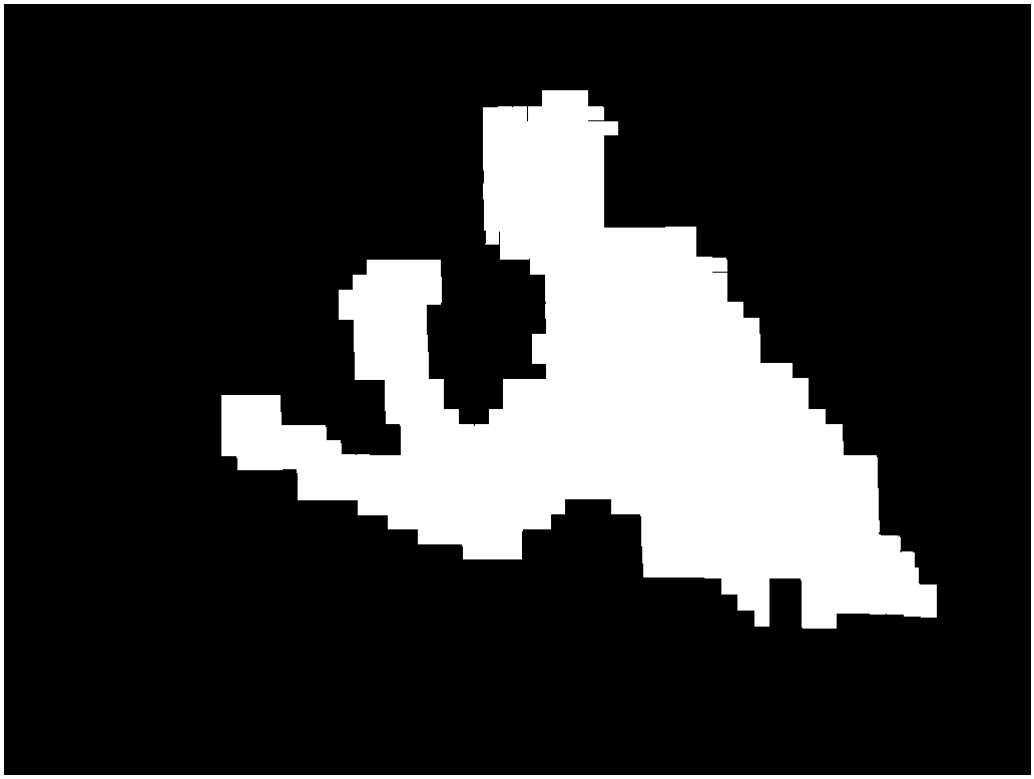


Figure 6.20: The result after treating $DF2$ with the median filtering scheme elaborated in Section 6.5.

Chapter 7

DISCUSSION, CONCLUSIONS AND FUTURE WORK

7.1 Discussion and conclusions

In this thesis autonomous scene classification models and methods based on similarity, motion and depth cues have been studied and developed to address the needs of scene activity tracking. It has been noted that, traditionally, motion based cues have been exploited to perform the task of foreground object segregation, fulfilling typical scene surveillance system requirements. In a standard motion based segmentation approach every current frame is subtracted from an estimated background frame that is usually generated either through the use of a temporal averaging process or the deployment of a learning rate based method. This helps in extracting the moving objects from the relatively static or slowly changing background. However, performance of this classical scene segmentation method is often jeopardised by the shadows cast by the moving objects that usually get segregated along with with the real targets from the background scene. In the first part of this thesis a computational model has been developed that can be used to label each pixel of a current frame as foreground, moving-shadow or background pixel to overcome the shadow related pitfall of a motion based segmentation method.

In Chapter 2 of this thesis it has been demonstrated how efficiently the computational model marks the moving-shadow pixels in a current frame scene. Indoor

scenes illuminated by a single fixed source of incandescent illumination have been chosen to illustrate how the model functions; the choice of such scenes has been made after noting their conspicuous non-appearance in the scene surveillance literature. Extensive statistical analysis of the model parameters has been undertaken and the results reported in the chapter to enable a user to choose the initial values of the required threshold values conveniently. A channel ratio test has also been developed in the same chapter that effectively reduces the false detection rate in cases where shadows are cast on neutral surfaces. However, it has also been observed that implementation of the pixel-wise moving shaded region search process with strict thresholds mainly marks the strong portion of the cast shadows. The fact that thresholds need to be relaxed for the inclusion of the soft portion of the shadow in the detected shadow region mask becomes apparent from the final results obtained after applying the shadow detection scheme in diverse indoor scenes containing both portions of shadows cast by foreground objects. However, any such attempt of detecting the entire shadow region by relaxing model threshold parameter values is usually accompanied by a rise in false detection rate. So, it can be deduced from the results reported in Chapter 2 that constraining the shadow search area is necessary to limit the false detection rate if model thresholds are to be relaxed to encompass the soft portion of a shadow along with its relatively stronger counterpart.

In Chapter 3 an attempt to achieve the result of detecting the entire moving shadow region without loosing control over the false detection rate has been made by using the computational model in tandem with a traditional motion-cue based scene segmentation method. Use of a standard background subtraction method helps in generating a binary mask containing a foreground object along with its shadow, if present; this constrains the shadow search area, thus making possible the relaxation of the thresholds of the computational model. The computational model is now capable of detecting and eliminating the entire false-target (shadow) regions in a scene. Restricting the shadow search area also speeds up the core shadow search process as the computational model is, otherwise, applied pixel-wise to label the foreground shaded pixels. The illustrated results obtained after applying the schemes, in tandem, clearly show that the idea of developing a combined approach has borne fruit.

It has also been noted while working on this combined approach that noise creeps into the binary mask generated from the difference frame due to improper choice of thresholds. Here it is to be kept in mind that an autonomous scene segmentation method imperatively requires an automatic selection of a threshold to develop a binary mask from a difference frame. Various popular outlier detection strategies have been studied and implemented in Chapter 3 to assess their suitability in generating an automatic threshold for the binary mask development process.
The performance of the combined method has also been compared with that of the original method in the same chapter through the use of various performance evaluation metrics; the results obtained suggest that the Hampel Identifier based outlier detection strategy can be conveniently used to generate a threshold automatically required for the binary mask development step. The computational model with relaxed thresholds can then be used subsequently to mark and eliminate nearly the whole portion of the foreground shadow regions.

In chapter 4 the developed computational model, without the channel ratio test, has been applied to segment foreground objects in recorded video streams of outdoor and indoor real-life scenes to assess its suitability both in terms of performance and associated time constraints. The complete version of the computational model has been deployed in the developed two-stage approach to observe and study every facet of its pixel-labelling capabilities. It is noted that though the method works well, in general, with real-time constraints, its performance sometimes falters resulting in false-object tracking and subsequently false alarms. The main reasons that jeopardise the functioning of the model include illumination-change and the fact that the size of the morphological filters, used in the process, are chosen on an *ad hoc* basis sometimes resulting in segmented blobs pertaining to the same region remaining disconnected. In the practical application described in the chapter, the method has been supplemented by an edge-map based histogram matching process to reduce the number of false alarms triggered by the process. It should be mentioned here that although this method makes the second stage of the approach, more or less, illumination change tolerant, restricted use of its capabilities should be made considering its computational intensiveness and the time constraints a real-time application carries.

In the second part of this thesis depth based cues have been exploited to perform the task of segregating foreground objects from the background scene. Out of the two classical ways of estimating depth or depth based cues, passive and active, the thesis focuses on a particular type of the latter to alleviate some of the intrinsic difficulties involved in the solving of the correspondence problem.

In Chapter 5 a practical mathematical model of a structured light projection based depth measuring arrangement has been developed. The model is capable of predicting accurate locations of spots, projected by a laser-diffractive-optical-element arrangement (LDA), on a flat screen placed at a particular distance from the set-up. Model equations can also be used to locate the projected locations of most of the structured pattern elements on the image plane of a camera, which is modeled as an optical sensor fitted with a thin lens viewing the scene under observation. The projections of a few rows on the peripheral regions of the pattern, lying on curves with high curvature values, could not be registered using the model equations. This is because of the fact that though the lens-system has been calibrated through the use of an off-the-shelf calibration technique to rectify the anomalies in the images viewed through it, the radial distortion of the lens is not fully compensated for. In terms of the use of the arrangement, the model equations can be used to find out the depth of each of the registered object points (structured pattern elements), predict how the spots will move on the image plane with the change of the distance of the object from the camera and, most significantly, in devising a method to increase the working volume of the system.

A method to calculate the depth of an object point from two successive corresponding projection location estimates has also been outlined in the chapter. This method usually proves to be useful as the planar co-ordinates of the centre of the image plane of the sensor does not, in general, coincide with that of the principal point of the lens, a problem that arises out of the camera parts assembly process. Finally, it should be mentioned here that the entire modeling shows how to increase the working volume of the system by giving the principal point of the camera similar orientation and specific rectilinear shifts with respect to the global co-ordinate system fixed on the LDA arrangement. This, in other words, means that the modeling allows an

increase in the operating volume of a typical structured light based range estimating camera without explicitly encoding the projection pattern, a method that usually brings additional requirements and constraints with it as discussed in Chapter 1 of the thesis.

In Chapter 6 it has been discussed how solely the structured spot position disparity information can be used to segment multiple non-overlapping or partially overlapping objects or a single object from the background. A method to segment multiple non-overlapping objects or a single object using the depth sensing arrangement, modeled in the previous chapter, has also been developed in Chapter 6. In the method developed, only spot position shifts on the camera image plane have been used to generate a difference frame containing the silhouettes of the objects. However, it has been noted that such difference frames, or subsequent binary masks generated using those, come out contaminated with noisy pixel clusters. Two median filtering schemes have also been developed in the same chapter to remove the noisy pixel clusters from the difference frames or the corresponding binary masks. In the first filtering scheme outlined, a difference frame is repeatedly filtered with the non-recursive version of a custom-made median filter, and finally a modified median filter is applied to generate the final result. Though the method yields satisfactory results, it is disadvantaged by the fact that two difficult thresholds have to be selected to ensure its proper functioning. In the second method, multistage unidirectional max/median and min/median filters have been used to de-noise the difference frames. Well understood $1-D$ median filter properties have been used to determine the size of the unidirectional sub-filters employed in the construction of the multistage filter. The results obtained after applying the method shows the scheme is not only capable of de-noising the binary frames but also of bridging the holes in the main object blobs. How the second filtering scheme can be realised using discrete morphological operators has also been elaborated in the same chapter. This has been done with the intention that a dedicated architecture built, following the the process described, can be used to speed-up the entire de-noising process.

7.2 Future Work

In this thesis a computational model has been developed to label every pixel in a current frame as a foreground, shadow or background pixel. Extensive statistical analysis has been conducted on the model parameters and results reported so that the initial guesses on the threshold values can me made conveniently when using the model to demarcate shaded regions in indoor scenes illuminated by a fixed incandescent source. More competent approaches employing robust statistics like median and median absolute deviation from the median (MAD), and mixture modeling using difference of Gaussians will be investigated in future to facilitate better understanding of the model and for better tuning of the model parameters.

The developed computational model has been combined with a classical motion based segmentation process in Chapter 3 of the thesis to overcome some of the difficulties associated with the individual approaches. This work can be extended to study the Fourier based methods of motion segmentation, thus assessing their suitability in real-life video surveillance applications.

In Chapter 4 of the thesis the computational model has been applied as the first stage of a two-stage approach to segment foreground objects from background scenes. Various learning rate based background update processes need to be studied and implemented to comprehend and appraise their effects on the performance of the computational model's pixel classifying capabilities.

In Chapter 5 a structured light based depth estimating arrangement has been modeled with the aim to isolate the foreground objects buried in some background scene using range and range based cues. It has been noted that the model equations cannot be used to register some of the object points lying on the peripheral regions of the pattern because lens distortion could not be fully compensated. In future, more complex lens calibration methods will be implemented to compensate for the distortions and, in turn, make full use of the depth sensing arrangement. In addition, experiments will be conducted to demonstrate the predicted increment of the working volume of the system that can be achieved without explicitly encoding the

structured pattern.

In Chapter 6 structured pattern disparity based information has been used to perform the task of foreground scene segmentation. Two median filtering schemes have also been proposed to denoise the difference frames containing the foreground scene, generated through the use of the developed active mechanism of scene classification. Out of the two median filtering approaches, the one employing multistage max/median and min/median filters and found to be more adequate for practical use, has been described using basic discrete morphological operators. Work is currently directed towards development of a dedicated architecture where careful use of the morphological operators will be made to achieve a speed-up of the median filtering scheme.

Finally, it should be mentioned that in this thesis mainly bottom-up approaches have been studied, developed and implemented to perform the task of automatic scene classification. In future, top-down approaches using correlation pattern recognition filters and deformable templates will also be studied to assess their suitability and performance in various video surveillance applications.

Bibliography

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992.

[2] J. C. Russ, *The Image Processing Handbook*, 5th ed. Canada: CRC Press, 2007.

[3] M. Basu, "Gaussian-based edge-detection methods- a survey," *IEEE Trans. on Systems, Man and Cybernatics*, vol. 32, no. 3, pp. 252–260, August 2002.

[4] W. K. Pratt, *Digital Image Processing*, 2nd ed. John Wiley and Sons Inc., 1991.

[5] R. Jain, "Dynamic scene analysis using pixel-based processes," *Computer*, vol. 14, no. 8, pp. 12–18, 1981.

[6] D. Vernon, *Fourier Vision*. Massachusetts, USA: Kluwer Academic Publishers, 2001.

[7] S. Ullman, *High-level Vision*. The MIT Press, 1996, provided in screen-viewable form.

[8] H. J. Caulfield and W. T. Maloney, "Improved discrimination in optical character recognition," *Appl. Opt.*, vol. 8, no. 11, pp. 2354–2356, 1969.

[9] C. F. Hester and D. Casasent, "Multivariant technique for multiclass pattern recognition," *Applied Optics*, vol. 19, pp. 1758 – 1761, 1980.

[10] B. V. K. Vijaya Kumar, "Minimum variance synthetic discriminant functions," *Journal of Optical Society of America A*, vol. 3, pp. 1579 – 1584, 1986.

[11] A. Mahalanobis, B. V. K. Vijaya Kumar, and D. Casasent, "Minimum average correlation energy filters," *Applied Optics*, vol. 26, pp. 3363 – 3640, 1986.

[12] R. Young, C. Chatwin, and S. B., "High speed hybrid optical/digital correlator system," *Optical Engineering*, vol. 32, pp. 2608 – 2615, 1993.

[13] A. Mahalanobis, B. V. K. Vijaya Kumar, S. Song, S. R. F. Sims, and J. F. Epperson, "Unconstrained correlation filters," *Applied Optics*, vol. 33, no. 17, pp. 3751 – 3759, June 1994.

[14] B. V. K. Vijaya Kumar, A. Mahalanobis, and R. D. Juday, *Correlation pattern recognition*. United Kingdom: Cambridge University Press, 2005.

[15] R. A. Kerekes and B. V. K. Vijaya Kumar, "Selecting a composite correlation filter design: a survey and comparative study," *Optical Engineering*, vol. 47, no. 6, pp. 067 202:1–10, 2008.

[16] M. Nixon and A. S. Aquado, *Feature Extraction and Image Processing*, 2nd ed. Hungary: Academic Press, 2007.

[17] W. Frei and C. Chen, "Fast boundary detection: A generalisation and a new algorithm," *IEEE Trans. Computers*, vol. C-26, no. 10, pp. 988 – 998, 1977.

[18] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780 – 785, July 1997.

[19] J. Ohya, J. Kurumisawa, R. Nakatsu, K. Ebihara, S. Iwasawa, D. Harwood, and T. Horprasert, "Virtual metamorphosis," *Multimedia, IEEE*, vol. 6, no. 2, pp. 29 –39, apr-jun 1999.

[20] I. Haritaoglu, D. Harwood, and L. Davis, "W4: real-time surveillance of people and their activities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 809 –830, aug 2000.

[21] T. Horprasert, D. Harwood, and D. L. S., "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. IEEE International Conference on Computer Vision ('99 FRAME-RATE Workshop)*, 1999.

[22] A. Prati, I. Mikic, and C. R., "Detecting moving shadows: Algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918–923, 2003.

[23] S. A. Shafer and T. Kanade, "Using shadows in finding surface orientations," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 145 – 176, 1983.

[24] Y.-T. Liow and T. Pavlidis, "Use of shadows for extracting buildings in aerial images," *Computer Vision, Graphics, and Image Processing*, vol. 49, no. 2, pp. 242 – 277, 1990.

[25] R. Irvin and J. McKeown, D.M., "Methods for exploiting the relationship between buildings and their shadows in aerial imagery," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 6, pp. 1564 –1575, nov/dec 1989.

[26] C. Jiang and M. Ward, "Shadow identification," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, jun 1992, pp. 606 –612.

[27] P. L. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," in *Proc. of the sixth British Machine Vision Conference*, 1995, pp. 347–356.

[28] J. Stauder, R. Mech, and J. Ostermann, "Detection of moving cast shadows for object segmentation," *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 65–76, 1999.

[29] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "The sakbot system for moving object detection and tracking," *Video Based Surveillance Systems-Computer Vision and Distributed Processing*, pp. 145 – 157, 2001.

[30] D. M. Hawkins, *Identification of Outliers*. Great Britain: Chapman and Hall, 1980.

[31] P. H. Menold, R. K. Pearson, and F. Allgower, "Online outlier detection and removal," in *Proc. 7th Mediterranean Conference on Control and Automation*, Haifa, Israel, June 1999, pp. 1110–1122.

[32] R. K. Pearson, "Outliers in process modeling and identification," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, January 2002.

[33] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[34] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, p. 252 Vol. 2.

[35] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *ICCV Frame Rate Workshop*, 1999.

[36] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, 10-13 2004, pp. 3099 – 3104 vol.4.

[37] M. Pic, L. Berthouze, and T. Kurita, "Adaptive background estimation : Computing a pixel-wise learning rate from local confidence and global correlation values (background estimation) (¡special section¿machine vision applications)," *IEICE transactions on information and systems*, vol. 87, no. 1, pp. 50–57, 2004-01-01. [Online]. Available: http://ci.nii.ac.jp/naid/110003213804/en/

[38] N. Alvertos, D. Brzakovic, and R. Gonzalez, "Camera geometries for image matching in 3-d machine vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 9, pp. 897 –915, sep 1989.

[39] J. Batlle, E. Mouaddib, and J. Salvi, "Recent progress in coded structured light as a technique to solve the correspondence problem: a survey," *Pattern Recognition*, vol. 31, no. 7, pp. 963 – 982, 1998.

[40] E. Hecht, *Optics*, 2nd ed. USA: Addison-Wesley Publishing Company, 1987.

[41] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330 – 1334, nov 2000.

[42] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[43] R. I. Hartley, "Theory and practice of projective rectification," *International journal of Computer Vision*, vol. 35, no. 2, pp. 115–127, November 1999.

[44] Y. Shirai and M. Suwa, "Recognition of polyhedrons with a range finder," in *Proc. Int. Joint Conference on Artificial Intelligence*, 1971, pp. 80 – 87.

[45] R. J. Popplestone, C. M. Brown, A. P. Ambler, and G. F. Crawford, "Forming models of plane-and-cylinder faceted bodies from light stripes," in *Proc. Int. Joint Conference on Artificial Intelligence*, 1975, pp. 664–668.

[46] G. Agin and T. Binford, "Computer description of curved objects," *Computers, IEEE Transactions on*, vol. C-25, no. 4, pp. 439 –449, april 1976.

[47] Y. Sato, H. Kitagawa, and H. Fujita, "Shape measurement of curved objects using multiple slit-ray projections," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-4, no. 6, pp. 641 –646, nov. 1982.

[48] O. Ozeki, T. Nakano, and S. Yamamoto, "Real-time range measurement device for three-dimensional object recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 4, pp. 550 –554, july 1986.

[49] Y. F. Wang, A. Mitiche, and J. K. Aggarwal, "Computation of surface orientation and structure of objects using grid coding," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 1, pp. 129 – 137, jan. 1987.

[50] M. Asada, H. Ichikawa, and S. Tsuji, "Determining surface orientation by projecting a stripe pattern," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 5, pp. 749 –754, sep 1988.

[51] G. Hu and G. Stockman, "3-d surface solution using structured light and constraint propagation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 4, pp. 390 –402, apr 1989.

[52] N. Shrikhande and G. Stockman, "Surface orientation from a projected grid," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 6, pp. 650 –655, jun 1989.

[53] Y. Wang, "Characterizing three-dimensional surface structures from visual images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, no. 1, pp. 52 –60, jan 1991.

[54] K. Kemmotsu and T. Kanade, "Uncertainty in object pose determination with three light-stripe range measurements," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 5, pp. 741 –747, oct 1995.

[55] R. Legarda-Saenz, T. Bothe, and W. P. Juptner, "Accurate procedure for the calibration of a structured light system," *Optical Engineering*, vol. 43, no. 2, pp. 464 – 471, feb 2004.

[56] M. D. Altschuler, B. R. Altschuler, and J. Taboada, "Laser electro-optic system for rapid (3-d) topographic mapping of surfaces," *Optical Engineering*, vol. 20, no. 6, pp. 953 – 961, 1981.

[57] E. Muller, "Fast three dimensional form measurement system," *Optical Engineering*, vol. 34, no. 9, pp. 2754 – 2756, 1995.

[58] B. Carrihill and R. Hummel, "Experiments with the intensity ratio depth sensor," *Computer Vision and Graphics Image Processing*, vol. 32, pp. 337 – 358, 1985.

[59] K. L. Boyer and A. C. Kak, "Color-encoded structured light for rapid active ranging," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 1, pp. 14 –28, jan. 1987.

[60] J. Le Moigne and A. Waxman, "Structured light patterns for robot mobility," *Robotics and Automation, IEEE Journal of*, vol. 4, no. 5, pp. 541 –548, oct 1988.

[61] H. Morita, K. Yajima, and S. Sakata, "Reconstruction of surfaces of 3−d objects by m-array pattern projection method," in *Proc. Int. Conf. on Computer Vision*, 1988, pp. 468 – 473.

[62] P. Vuylsteke and A. Oosterlinck, "Range image acquisition with a single binary-encoded light pattern," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 2, pp. 148 –164, feb 1990.

[63] Z. J. Geng, "Rainbow three-dimensional camera: new concept of high speed three-dimensional vision system," *Optical Engineering*, vol. 35, no. 2, pp. 376 – 383, 1996.

[64] C. Wust and D. W. Capson, "Surface profile measurement using colour fringe projection," *Machine Vision and Applications*, vol. 4, no. 3, pp. 193 – 203, 1991.

[65] P. M. Griffin, L. S. Narasimhan, and S. R. Yee, "Generation of uniquely encoded light patterns for range data acquisition," *Pattern Recognition*, vol. 25, no. 6, pp. 609 – 616, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/B6V14-48MPP18-1MT/2/1a4388244b5925960c88e39f06d350c7

[66] M. Maruyama and S. Abe, "Range sensing by projecting multiple slits with random cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 6, pp. 647 –651, jun 1993.

[67] A. Sanderson, L. Weiss, and S. Nayar, "Structured highlight inspection of specular surfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 1, pp. 44 –55, jan 1988.

[68] C. Guan, L. Hassebrook, and D. Lau, "Composite structured light pattern for three-dimensional video," *Opt. Express*, vol. 11, no. 5, pp. 406–417, 2003. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-11-5-406

[69] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: A unifying framework for depth from triangulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 296–302, 2005.

[70] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1079–1087, 2004.

[71] G. J. Hahn and S. S. Shapiro, *Statistical Models in Engineering*. USA: John Wiley and Sons Inc., 1967.

[72] *MathWorks Inc.*, 3 Apple Hill Drive, Natick, MA 01760-2098, USA.

[73] M. H. DeGroot and M. J. Schervish, *Probability and Statistics*, 3rd ed. USA: Addison-Wesley, 2002.

[74] B. K. Mitra, R. Young, and C. Chatwin, "On shadow elimination after moving region segmentation based on different threshold selection strategies," *Optics and Lasers in Engineering*, vol. 45, no. 11, pp. 1088–1093, November 2007.

[75] B. K. Mitra, M. K. Fiaz, I. Kypraios, P. M. Birch, R. Young, and C. R. Chatwin, "Performance analysis of a modified moving shadow elimination method developed for indoor scene activity tracking," in *SPIE Europe Security + Defence- Optics and Photonics for Counterterrorism and Crime Fighting - IV*, vol. 7119, Cardiff, Wales, United Kingdom, September 2008, pp. 71 190A–1:10.

[76] R. K. Pearson, *Mining Imperfect Data: Dealing with Contamination and Incomplete Records*. Philadelphia: SIAM, 2005.

[77] K. Onuguchi, "Shadow elimination method for moving object detection," in *Proc. International Conference on Pattern Recognition 1*, 1998, pp. 583–587.

[78] S.-N. Lim and L. Davis, "A one-threshold algorithm for detecting abandoned packages under severe occlusions using a single camera," University of Maryland, College Park, CS Dept., Tech. Rep. CS-TR-4784, February 2006.

[79] F. Porikli, "Detection of temporarily static regions by processing video at different frame rates," in *Proc. IEEE International Conference on Advanced Video and Signal-Based Surveillance*, London,UK, September 2007.

[80] Y.-l. Tian, R. Feris, and A. Hampapur, "Real-time detection of abandoned and removed objects in complex environments," in *Proc. IEEE International Workshop on Video Surveillance in conjunction with ECCV'08*, Marseille, France, 2008.

[81] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier, "Left-luggage detection using homographies and simple heuristics," in *PETS*, 2006, pp. 51–58.

[82] S. Guler, J. A. Silverstein, and I. H. Pushee, "Stationary objects in multiple object tracking," in *Proc. IEEE International Conference on Advanced Video and Signal-Based Surveillance*, London, UK, September 2007.

[83] P. T. N. Krahnstoever, T. Sebastian, A. Perera, and R. Collins, "Multiview detection and tracking of travellers and luggage in mass transit environments," in *PETS*, 2006, pp. 67–74.

[84] J. M. del Rincn, J. E. Herrero-Jaraba, J. R. Gomez, and C. Orrite-Urunuela, "Automatic left luggage detection and tracking using multiple cameras," in *PETS*, 2006, pp. 59–67.

[85] K. Smith, P. Quelhas, and D. Gatica-Perez, "Detecting abandoned luggage items in public space," in *PETS*, 2006, pp. 75–82.

[86] B. K. Mitra, P. Birch, I. Kypraios, R. Young, and C. Chatwin, "On a method to eliminate moving shadows from video sequences," in *Proc. SPIE Photonics Europe- Optical and Digital Image Processing*, vol. 7000, Strasbourg, France, April 2008, pp. 700 012–1:9.

[87] G. Bradski and A. Kaehler, *Learning Open CV*. USA: O'Reilly, 2008.

[88] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters," *IEEE Signal Processing Letters*, vol. 2, no. 11, pp. 360–363, November 2002.

[89] B. McCane, "Edge detection," Department of Computer Science, University of Otago, Dunedin, New Zealand, Tech. Rep., February 2001.

[90] P. R. Bevington and D. K. Robinson, *Data Reduction And Error Analysis For The Physical Sciences*, 3rd ed. New York, USA: McGrawHill, 2003.

[91] J. R. Taylor, *An Introduction to Error Analysis*. Mill Valley, California: University Science Books, 1982.

[92] K. J. Gasvik, *Optical Metrology*, 2nd ed. Great Britain: John Wiley and Sons, 1995.

[93] S. G. Tyan, *Two Dimensional Digital Signal Processing II*, ser. Topics in Applied Physics, T. S. Huang, Ed. Germany: Springer-Verlag, 1981.

[94] N. C. Gallagher, Jr and G. L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, no. 6, pp. 1136–1141, December 1981.

[95] T. A. Nodes and N. C. Gallagher, Jr, "Median filters: Some modifications and their properties," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-30, no. 5, pp. 739–746, October 1982.

[96] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. PAMI-9, no. 1, pp. 74–90, January 1987.

[97] G. R. Arce and M. P. McLoughlin, "Theorectical analysis of the max/median filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, no. 1, pp. 60–69, January 1987.

[98] G. R. Arce and R. E. Foster, "Detail-preserving ranked-order based filters for image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 1, pp. 83–98, January 1989.

[99] P. A. Maragos and R. W. Schafer, "A unification of linear, median, order statistics and morphological filters under mathematical morphology," in *Proc. IEEE ICASSP*, March 1985.

[100] P. A. Maragos and R. Schafer, "Morphological filters — part ii: Their relations to median, order-statistic, and stack filters," *IEEE Trans. Acoustic, Speech and Signal Processing*, vol. 35, no. 8, pp. 1170 – 1184, 1987.

[101] P. Soille, "On morphological operators based on rank filters," *Pattern Recognition*, vol. 35, no. 2, pp. 527–535, February 2002.